QRPp

Journal of the Northern California QRP Club
Volume IV, Number 4, December 1996

The "38 Special"Transceiver 3
Ori Mizrahi-Shalom, AC6AN
Mint Tin Rainbow SWR Bridge and Tuner 8
Joe Everhart, N2CX
Homebrewing a Software Keyer 14
Daniel Wee, 9V1ZV
Arizona ScQRPion Mini-Mag Paddles 31
Floyd Smithberg, NQ7X
A Mighty Fine Tuner 37
John Spoonhower, KC2DU
Fix the Thump in Your NW80-20 Series Rigs 38
Paul Harden, NA5N
Huff & Puff VFO for the Epiphyte 40
Ed Burke, KI7KW
Casecade Steel Case Crystal Mod 44
Dave Meacham, W6EMD
A Piezo-Electric Key 44
Glen Leinweber, VE3DNL
Sierra Power Output - Revisited 47
Bob Follett, WA7FCU
Pacificon 96: The First Annual West Coast QRP Symposium 49
Doug Hendricks, KI6DS
Figuring Sheet Metal Bend Allowances 58
Vic Black, AB6SO
Putting the KC-2 in the Cascade 59
Dave Meacham, W6EMD
XIT Mod for the Sierra 60
Eric Swartz, WA6HHQ
Convert the 49er to a 39er 61
Jeff Anderson, WA6AHL
FYBO QRP Contest Announcement 61
Joe Gervais, KC6NEV
QRP Hints & Kinks 63
Paul Harden, NA5N

From the Editor

by Doug Hendricks, KI6DS
862 Frank Ave.
Dos Palos, CA 93620
Internet: ki6ds@telis.org
Phone: 209.392.3522

Greetings again from the editor. This is the end of the 1996 QRP year, and what an exciting year it has been. We saw the 49er, KC-2, St. Louis Tuner, St. Louis Vertical, and with this issue, the "38 Special", all NorCal kits.

This issue also features an article with lots of pictures from the West Coast QRP Symposium that was sponsored by NorCal and held in conjunction with Pacificon 96 at Concord, California. We had a wonderful time and hope that you can join us next year for the second annual event. Plan on keeping the third weekend of October open, as we plan on meeting in Concord once again.

This year also was a sad one for Jim and I. Jim lost his lovely wife Electra, to cancer in June. He received many, many messages and cards of condolence from his friends and wishes to say thank you. My wife's father died in late October from Alzheimer's Disease, and again I want to echo Jim's thank you to all of you who sent messages and cards. They were appreciated very much by both Jim and my wife JoAnne and me.

This issue features the "38 Special" and the Rainbow Tuner, the two co-winners of the NorCal sponsored Dayton Building Contest Design Contest. The purpose of the design contest was to encourage the development of cheap, easy to build projects, and boy do we have two great winners. Joe Everhart, N2CX, from the New Jersey QRP Club and a long time NorCal QRP Club member, has an outstanding project with his Rainbow Tuner. This one will be kitted by the guys from the New Jersey QRP Club. It was not possible for NorCal to kit both projects, and the New Jersey QRP Club stepped forward to offer their services which we appreciate very much. It is nice to have another club producing a kit for QRPers to build. I strongly encourage you to support their efforts.

Ori Mizrahi-Shalom, AC6AN did a lot of the work with the 49er mods that appeared in the June issue. He loves the idea of a simple radio, that is cheap to build, and encourages experimentation. So, he has designed the "38 Special", which will be kitted by NorCal. The "38 Special" has a superhet receiver, sidetone, offset, and a tuning range of 25 kHz. Plus, the circuit board has been designed to allow for adding an RIT, improved IF filter, a simple mod to increase the power to 3 Watts, and there will even be space for the new TiCK keyer that has been introduced by Embedded Research. Complete details for the mods are in the kit manual. Congratulations to Ori and Joe for their wonderful work.

Wayne Burdick is busy working out the final details on his new design, the SST, which stands for the "Super Simple Transceiver". NorCal will kit this rig, which will be available in late February. Complete details will be in the next issue. Wayne had hoped to have it ready by this issue, but just could not get it finished in time. We did not want to rush the project, and want it to be right, the first time. The SST will be available in two forms, as a bare bones kit, with PC Board and parts only, and as a full kit with case, connectors and controls. Pricing details are not firm at this time.

Finally, starting with this issue, we will have a Winter, Spring, Summer and Fall issue We will try to publish on the same schedule as before. 72, Doug, KI6DS

The "38 Special" Transceiver

by Ori Mizrahi-Shalom, AC6AN
2841 Burdick Way
San Jose, CA 95148
e-mail: ori@juno.com

What radio??

1) is a superhet with only two NE602s

2) has no discrete transistors

3) sells for $25

Can't guess? The answer is the "38 Special". The "38 Special" is an exercise in minimization. We feel that it is "special" because of the following:

*It works on the under-utilized 30M band

*Runs of f a 8V supply (due to limitations of some components)

*Special in its architecture

*Excellent price/performance ratio

*Last, but certainly not least, the fact that it offers a great platform for another year of hacking...

The "38 Special" is the result of an effort which started with hacking the 40-9er transceiver. A novelty radio, the direct-conversion 40-9er was quite limited, especially for weak signal work. The "38 Special" rectifies most of these problems and even adds a few nice features. It came to life due to the design contest challenge and the never ending enthusiasm and encouragement by Doug, KI6DS, who also came up with the great name.

GENERAL DESCRIPTION

The "38 Special" is a superhet transceiver for the 30M band. The first thing you'll notice looking at the schematics is the lack of discrete transistors. I have been playing with this concept for some time and the 38 Special was the right vehicle. The transmit section relies on a TTL buffer. I extended this concept to utilize the same TTL chip for other tasks. Although it's an NE602-based superhet transceiver, the "38 Special" incorporates only two NE602s. I reuse the product-detector as the transmit mixer by channeling different signals to that chip on receive and transmit. Also, the traditional LM380/386 is gone in favor of a more versatile dual op-amp circuit for the audio section.

RECEIVER FRONT-END

The receiver front-end starts with a back-to-back diode switch. The "38 Special" utilizes 1N4007 diodes for the switch. This diode has a PIN structure and provides low insertion loss, although it suffers from poor zero bias isolation and does not offer a strong IMD performance like an RF-rated PIN diode. [1] It is superior to the 1N914 or similar diodes in this type of design. Next is a toroidal impedance transformer with a 10.1 MHz tuned circuit at its output, providing additional front-end selectivity to that offered by the transmit output network. Provisions were made to include a 10 KOhm pot for RF-GAIN control. The 10.1 MHz RF signal is fed to the input of the NE602 receive mixer, where it is mixed with the 22.1 MHz VXO to generate the 12.0 MHz IF frequency.

THE VXO

The superhet circuit enabled me to use standard crystals and avoid the high price of custom ham-band crystals. Many crystal combinations work for most HF bands. I chose a high frequency first crystal to achieve a high frequency swing. [2] This required a relatively high IF in the simple receiver. The NE602 Collpits oscillator required a high DC bias for a large swing, provided by a 3.9 KOhm resistor at pin 7. Although well below the value recommended, this resistor provides for stable operation of the NE602. "Rubbering" the crystal with a varicap allows relocating the tuning pot away from the oscillator, if desired. A 1N4000-family diode works here nicely as a varicap. [3]

A little assist from a molded inductor yields a tuning range of 25 KHz. A

little hint here for the experimenters, do not replace this inductor with a toroid. The low Q helps to increase the pulling range. The VXO signal is mixed inside the NE602 with the received signal to produce an IF output of 12.0 MHz, which is the difference between the VXO and the RF frequency. During transmit, a 22.1 MHz signal is taken from the Collpits oscillator and injected into the input of the transmit mixer.

IF FILTER

The IF filter is implemented with a single crystal. It is a few KHz wide, due to budget constraints. Provisions for a better IF filter are included in the board layout. The main selectivity is achieved at the audio stage and the wide IF filter greatly simplifies the alignment of the receiver. The wide IF filter provides very little "wrong" sideband attenuation. Due to this, the "38 Special" in the stock form cannot be classified as a single-signal receiver. It will take a much sharper IF filter to achieve that. Although you will hear the same signal twice, the sharp audio filter totally eliminates the "off" signal.

PRODUCT DETECTOR

In the cost cutting tradition I left out the "traditional" third NE602 for the transmit mixing. Instead, the 38 SPECIAL reuses the product-detector for the same function. On receive, an oscillator (IF frequency) is mixed with the IF signal and it results in a low-level audio signal. On transmit, a signal from the VXO is mixed with the IF frequency oscillator in the second NE602. The selection of the input signal to the second NE602 is done by means of a 4066 analog multiplexer. Other than the switching of signals with the 4066, the receiver is similar in concept to most NE602-based superhet rigs.

RECEIVE OFFSET

Sharing the product-detector and transmit mixer required a "trick" to achieve a receive offset. The 38 SPECIAL "pulls" the IF frequency oscillator about 500 Hz up on receive with a 100 pF capacitor in series with the 12.0 MHz oscillator crystal. During transmit, this capacitor is shunted to ground with a parallel forward-conducting diode, so the crystal oscillates right on its fundamental frequency, resulting in a zero-beat transmit signal. The down-conversion at the first mixer and this oscillator pulling up on receive combine to yield the "right" receive sideband at a lower frequency. So, although the IF filter allows either sideband through, it is easy to identify the "right" one. This is not an issue when you call a CQ. The answering station is on the right frequency, if it zero-beats with your transmit signal.

AUDIO AMP/FILTER

The audio is filtered and amplified by an NE5532A dual op-amp, instead of the "traditional" LM380 or LM386 chips. [4] The 5532 requires more external components, but it gives a higher gain, and more important, the circuit also forms a sharp band-pass filter. From that point of view, the "38 Special" is superior to most NE602-based radios. This amp delivers about 60 dB of gain while driving a walkman-style headphones. The filter offers a 50 Hz -6 dB bandwidth and about 400 Hz at -30 dB. This circuit uses a dozen more components than an LM386. But they are probably the most cost-effective components in the whole radio!

TRANSMIT CHAIN

As mentioned before, the product-detector doubles as a transmit mixer. The signal on the output is filtered by a tuned circuit, and the 10.1 MHz output of 100 mV is used to drive the two-stage transmit amplifier. A few examples in the literature describe the use of TTL chips (and other logic devices) for a low-power transmitter. [5]

The most interesting article on the subject was written by Len Smith and appeared in QST. Len used an octal inverting buffer with eight individual active devices, but really utilized only five of them. This circuit gave me the idea of using the

leftover devices for sidetone generation and other tasks.

One of the inverters is biased as a high gain linear amplifier. The 100 mV at its input comes out as a few volts on swing at the output. The single inverter is strong enough to drive the final circuit, which is made of four parallel inverters. Depending on the output matching, this circuit can deliver well over half a watt of output power. I chose to leave it at 400 mW for the sake of cool and safe operation of the final. The board supports additional circuitry (not supplied with the kit) for a higher output of up to 5W.

THE OUTPUT NETWORK

The use of only 8V supply required very low impedance for the final to give any appreciable output power. The matching is easily done by an L-C-L-C type network. [6]

SIDETONE

Two inverters combine in a simple oscillator circuit, as described in the ARRL handbook. The oscillator is clamped to ground during receive but is free-running during key-down. The output is attenuated by means of a large series resistor. There was no need to filter the square waveform, as the audio bandpass circuit does that anyway.

ADDITIONAL LOGIC

One inverter of the octal buffer chip forms the receive/transmit logic. It inverts the logic state of the "key" line, so when key is down, this signal is at full Vcc and when key is open, this signal is at 0V. The availability of the T and R signals simplifies the implementation of the T/R circuitry.

ALIGNMENT

The "38 Special" was designed with a novice builder in mind. There are only two alignment steps, although more parameters could be tweaked by the experienced builder. The receiver alignment consists of peaking the front-end trimcap for the highest receive audio.

The transmitter alignment is a bit more tricky. The radio transmits 10.1 MHz signal, when it has a 12.0 MHz IF signal in the output of the transmit mixer. This requires care when tuning the transmit filter trimcap. One way of doing that is by listening to your signal on another receiver. Tune close to your center frequency and look for the adjustment that results in an output with the least close-in spurs. The two alignment steps do not require any test equipment, although having such equipment can improve the alignment.

THE NEXT STEP

The final design reflects cost-cutting and other changes to simplify the alignment and kitting of the design. It is a superhet with offset and sidetone at a 40-9er price! As such, it has limitations, of course. I look at it as a product and a development platform. There will be many that will assemble the basic unit and have lots of fun with it in the stock form. But there will be those who want to do things their way. This radio was designed for both.

I will not continue its development. I leave that to the hackers and tinkerers out there. The "38 Special" was designed to continue the tradition that started with the 40-9er - mods by the dozen. There are many possibilities. In fact, some mods are being developed as we speak. I certainly encourage people to do just that. All I ask is that you share with the QRP community and let us all know of your adventures...

For those that build it stock or custom, plain or modified, hot or low key - have fun. I can't wait to see the entries for Dayton and Pacificon next year. I also hope to get more activity on the 30M band. Maybe we will finally know what the propagation properties really are on 30...

ACKNOWLEDGEMENTS

Again, many thanks to Doug Hendricks for the encouragement. Special thanks to Dave Fifield, KQ6FR, who built the second prototype and beat me to the first QSO - 549 on 200 mW from San Jose

to Spokane! Dave is a master builder, both of circuits and enclosures. He came up with many additions and suggestions that made this radio nicer and more robust. Special thanks to the XYL, who became a "radio-widow" during the last month...

NOTES:

(1) QST, Dec 1994 pp. 25-27.

(2) A similar concept is used in the Mizuho MX7S radio.

(3) See Jim Pepper's Deluxe QRP station, although not for rubbering a crystal.

(4) One designed by NN1G uses the 5532A for an audio amp but the filter seems to be not as sharp (that radio has a sharp IF filter)

(5) QRPp Mar. 1994 p. 58 and W1FB's Design Notebook p. 156.

(6) Solid State Design p. 53.

"38 SPECIAL" Parts List

C1,C2,C4,C6,C18,C23, C3,C10	0.01 uF 150
C5,C12,C22,C27,C28, C29,C30,C33, 36 C37,C38,C40,C43	0.1 uF
C35	0.001uF
C7,C8,C9,C11,C13,C14	47
C15,C19	100
C16,C34,C39	22uF/16V Elect.
C17,C20,C31	5
C21,C24	220
C25	1300 SM 5%
C26	560 SM 5%
C32	220uF/ 16V Elect.
D1,D2	1N4007
D3	1N4004
D4,D5	1N914
L1	4.7 uH (molded)
L2	T37-2,21T
L3	T37-2, 8T
L4	T37-2,12T
P1	100K pot*.
P2	1K pot.*
R1,R2,R3,R14	1.3K
R4,R6,R7	10K
R5	3.9K
R8	2.2K
R9,R16	270K
R10	130K
R11	2.2 MEG
R12,R13,R19	30K
R14	4.7K
R18	1.2K
R17	1K
T1	T37-2; 2TP:S18T
TC1,TC2	5-60 pF trimcap
U1,U3	NE602AN
U2	CD4066
U4	74HC240
U5	NE5532A
VR1	7808CT
X1	22.1184 MHz Xtal
X2,X3	12.0 MHz Xtal (matched)

*Off board component, not in kit

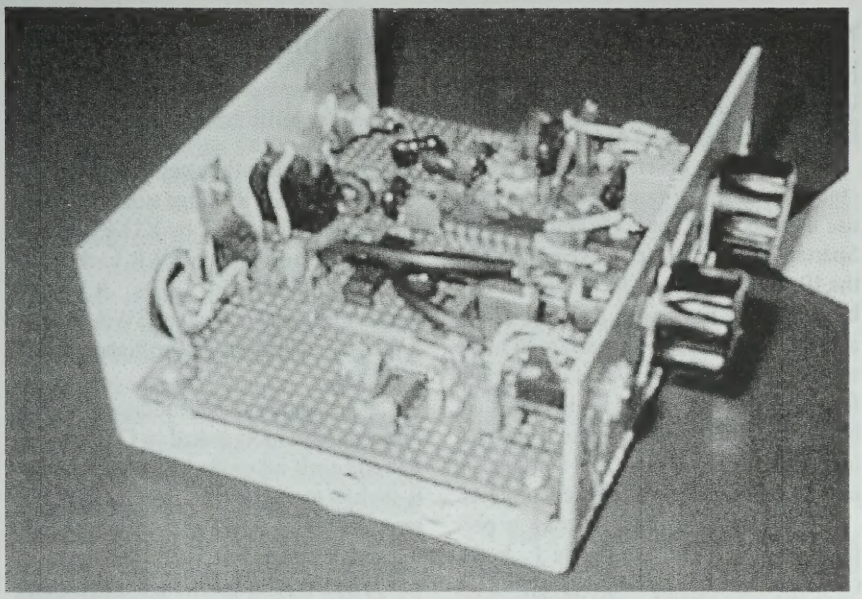
All caps are disk ceramic unless otherwise noted. All caps are in pF unless otherwise noted. All caps are 10% tolerance unless otherwise noted. All resistors are 1/4W, 10%. The schematic for the "38 Special" appears in the centerfold section of the magazine.

KITS

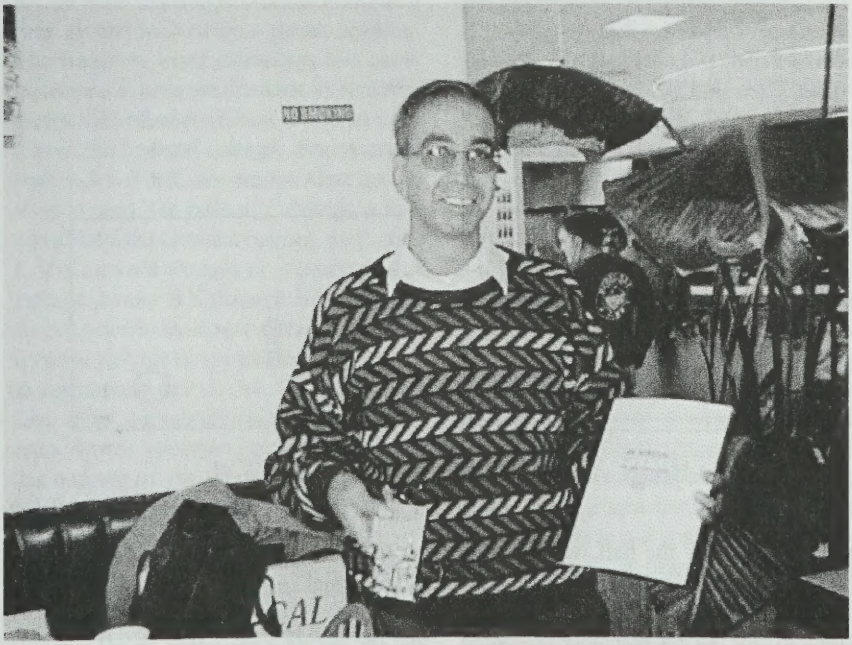
Parts Kits for the "38 Special" are available from:

Jim Cates
3241 Eastwood Rd.
Sacramento, CA 95821

Cost for the kit is $25 for the pcboard, board mounted parts, and instruction manual. Kit does not include connectors, controls, or case. Shipping and handling is $3 for US addresses, $5 for outside of US addresses. US funds only. Make checks or money orders out to Jim Cates, NOT NorCal.



Prototype of the "38 Special"



Ori Mizrahi-Shalom, AC6AN, designer of the "38 Special" with the prototype and his notes.

Mint Tin Rainbow SWR Bridge and Tuner

by Joe Everhart, N2CX
work: jeverhart@cayman.vf.mmc.co
home: n2cx@voicenet.com

The circuit described in the article really doesn't tune rainbows, but it does use a rainbow of sorts. It embodies the Spartan design philosphy exemplified by the 40-9er Transceiver. It combines a very basic antenna tuner with a simple-to-use accurate SWR meter. Intended to fit in the same type tin as the 40-9er, it can be built for under $25.00. As will be described below, the heart of the Rainbow Tuner is a user-friendly SWR indicator that can be used by itself in other homebrew projects.

The antenna tuning function is performed by L1 and C7 in Figure 1. In the configuration shown, they are connected as a parallel tuned resonant circuit with a tapped output. It is intended for use with a half wave end fed wire. The high impedance presented by the half wave wire is transformed to 50 ohms by taps on the toroidal coil. A compression mica trimmer is used to tune out any reactance presented by the antenna and inductor taps are selectable to give a close match to 50 ohm feedline. More detail will be given later about antenna length selection and tuneup.

A novel SWR bridge was written up in the June 1995 QST by K1KP. It used the familiar toroid type SWR bridge and replaced a meter with LEDs. It had no means of displaying actual SWR, merely a relative indication via brilliance of the its LEDs.

The Rainbow SWR indicator uses a bridge circuit more suited to QRP operation and a self-adjusting LED indicator that is slightly more complex than K1KP's but provides both relative indications for tuning and an exact final SWR reading.

The SWR bridge in Figure 1 consists of resistors R1, R2 and R3. R1 and R2 form a voltage divider with exactly half the input voltage present at their middle. This is rectified by D1 to produce a dc voltage proportional to forward RF power. The other half of the bridge is comprised of R3 and the output load, either an antenna or the tuner circuit, whichever is connected. When the load is 50 ohms, the voltage at the junction of R3 and the load is the same

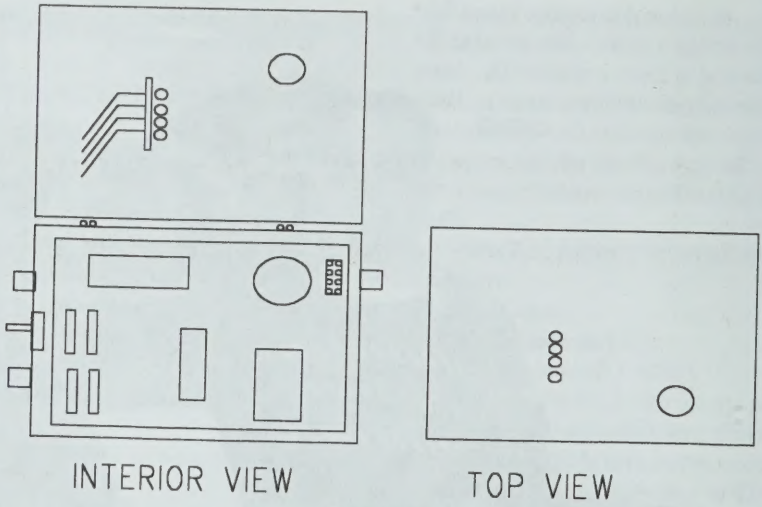


FIGURE 1

as at the R1/R2 junction. This corresponds to a 1:1 SWR and the voltage difference across the bridge arms is zero. When the load is not 50 ohms the difference voltage is proportional to the SWR. D2 rectifies the difference voltage to provide the reflected voltage.

Using resistors for the SWR bridge provides a real advantage when the Rainbow tuner is used with simple QRP rigs. Most SWR indicators pass their output SWR right on through to the rig driving them. Transmitters like the one used in the 40-9er and others misbehave with high SWR loads. At best, they become unstable and may oscillate generating off-frequency spurious signals. At worst, a high SWR load may destroy the final transistor. The absorptive bridge in the Rainbow tuner limits SWR that the transmitter sees to 2:1 maximum.

Most SWR indicators require the operator to adjust a meter or other indicator for full scale reading on forward power and then switch to a reverse power reading. The Rainbow tuner eliminates this complication. It uses the fact that in an SWR bridge, the reflected voltage is a fixed fraction of the forward voltage. For example with a 3:1 SWR, no matter what power level is used, the reflected voltage is *always* half the forward voltage. In Figure 1, the forward sample is connected to a voltage divider RX through RZ. The resistors are chosen to set a fraction of the forward voltage at comparator U1 inputs to correspond to 5:1, 3:1, 2:1 and 1.5:1. The other comparator inputs are fed directly by the reflected voltage. LEDS at the outputs of the comparators indicate SWR by lighting in response to the compared forward and reflected voltages.

The LEDs form the rainbow display. As shown in Table I, only the green LED is on only for SWR less than 1.5:1, both the green and yellow LEDs light with SWR between 1.5:1 and 2:1 and so on. Two levels of intensity are provided on the red LED for the highest SWR because I ran out of colors for cheap LEDs! No voltage regulator is needed by the comparator since it relies only on the resistive divider for accuracy.

Table I: Rainbow Display Interpretation

SWR	RED	ORG	YEL	GRN
> 5:1	O+	O	O	O
> 3:1	O	O	O	O
> 2:1	X	O	O	O
> 1.5:1	X	X	O	O
< 1.5:1	X	X	X	O

Note: O means LED is ON, O+ is higher intensity for RED LED. X means LED is OFF.

The SWR indicator is switched in or out of the circuit by S1. It is not connected at all times for two reasons. First, the tuner has a 6 dB loss even with a good match to the antenna. And secondly, power is required for the LEDs when the indicator is being used.

The tuner SWR indicator senses RF and turns itself on with less than 150 mW of RF. Q1 is a special MOSFET with a very low turn on voltage. I've measured a number of them and all have turned on with less than 1.5 volts.

The recommended battery, B1 has a rated milliampere hour capacity of only 33 mAH and the indicators draws about 10 to 50 mA, depending on how many are lighted. So battery life is prolonged by using the indicator only when needed. Longer life can be obtained with a 9-volt alkaline battery, but it won't fit into a a mint tin along with the tuner circuit board.

The tuner is sized to fit in a British mint tin with room for the SWR bridge selection switch and some small connectors - I used RCA phono jacks for their small size. The bridge has its own in and out jacks which can be connected to the tuner section with a short coax jumper. This way the tuner and SWR bridge can be used separately. If you want, you can

use a small switch to go between the bridge and tuner sections and eliminate the extra connector and coax lead.

Figure 2 is a sketch of the tuner as installed in the tin. The printed circuit board fits into the tin and is insulated from the metal box by a piece of cardboard beneath it. The battery is installed in a holder for a type N cell. Access to the tuner variable capacitor is via a hole punched in the lid. 6-32 screws and knurled nuts are used for antenna wire and counterpoise board and wires soldered to them to go off-board.

Operation is very simple. To use just the SWR bridge, attach the rig and antenna cables to the proper connectors and set the switch to the "in-line" position. As soon as you transmit, the circuit will turn itself on and you can read the SWR using the LEDs. When the RF disappears the tuner shuts off dc power. Set the switch to "bypass".

The antenna tuner is intended for a

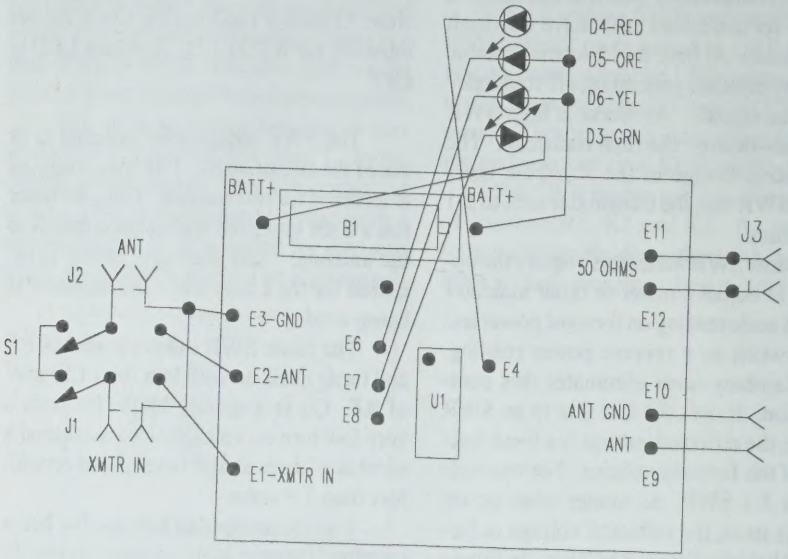


FIGURE 2

conections. Shoulder washers insulate the "hot" antenna wire hardware.

The display LEDs are plugged into a SIP (single inline socket) strip cut down to 8 pins. The LED leads are bent at right angles so that the socket can be glued inside the mint tin lid and the LEDs are visible through holes in the tin lid. A short piece of ribbon cable is soldered to the SIP socket and run to the printed circuit connections. I prefer to use pins in pc boards to connect extrnal wires so pins like the Vector T-44 types are soldered to the pc

half-wave end-fed wire. An antenna of this type presents an impedance of 1000 to 10000 ohms or so. To check out the tuner and learn how it works, connect a non-inductive resistor in that range across the antenna and ground terminals and a coaxial jumper from the tuner phono jack to the SWR bridge ANTENNA jack. Use a 40 meter rig with an ouput of 150 mW to 1 watt to feed the RIG jack. Set the tuner tap jumper between between the LOW Z taps.

Set the switch to the IN-LINE

postition and keyup the transmitter while monitoring the LEDS. Adjust the tuner variable capacitor with a non-metallic tuning tool for lowest SWR. If it isn't below 1.5:1, try the next highest tap and repeat the tuning. You should be able to find a tap and tuning setting which lights only the green LED.

Tuning with an antenna is done the same way. Use a single wire about 67 feet long and at least one 33 foot counterpoise. Try to get the antenna wire at least 15 to 20 feet off the ground. Key up your transmitter and tune as you did with the resistor. Due to the tuner's limited adjustment range you may find it necessary to trim the antenna slightly to get minimum SWR. It is recommended that once you have found a wire length that tunes up well, always use that wire and counterpoise.

Parts List

B1	12V lighter battery, RS
B2	"N" battery holder, RS
C1,C2,C5,C6	.05 uF mono cap, DK
C3, C4	1000 pF mono cap, DK
C7	150pF mica trimmer, DSP
D1, D2	1N34, RS
D3	Green 3 mm LED, DK
D4	Red 3 mm LED, DK
D5	Orange 3 mm LED, DK
D6	Yellow 3 mm LED, DK
L1	T68-2, DSP
Q1	VN10KM,DSP
Q2	2N3906 Various
R1, R2, R3	51ohm 5%,1/2W, DK
R4, R5	10K,1%, 1/4 W, DK
R6	56.2K, 1%, 1/4 W, DK
R7, R8	34K, 1%, 1/4 W, DK
R9	26.1K, 1%, 1/4 W, DK
R10	40.2K, 1%, 1/4 W, DK
R11	10K, 5%, 1/4 W, DK
R12	15K, 5%, 1/4 W, DK
R13, R16, R17	1K, 5%, 1/4 W, DK
R14, R15,	2K, 5%, 1/4 W, DK
S1	Mini toggle switch, RS
U1	LM339, DK or RS
15	T-44 pins, DK
3	RCA jacks, DK
40"	28 ga. magnet wire
3"	ribbon cable
1	4 pos X 2 0.1 spaced terminal block
1	8 pos SIP socket, RS

Misc. hardware

The parts shown are for maximum accuracy and performance. However some substitutions can be made with more avaialable components. Resistors R1, R2 and R3 directly affect bridge accuracy and power rating. 47 ohm 5% values can be used with only a slight effect on accuracy (the prototype used them!). The 1/2 watt values shown will be adequate for intermittent use with transmitter power up to 2 watts. Smaller wattage resistors will burn up at that power level. Of course 1 watt values will handle up to 4 watts and 2 watters theoretically can handle 8 watts. Prolonged transmit periods will cause excess component heating.

1% resistors are shown for R6 through R10 and R18 in the comparator circuit. 5% values close to these can be substitued but will result in degraded SWR accuracy. Besides, the Digikey resistors are cheap - only 54 cents for 5 of them.

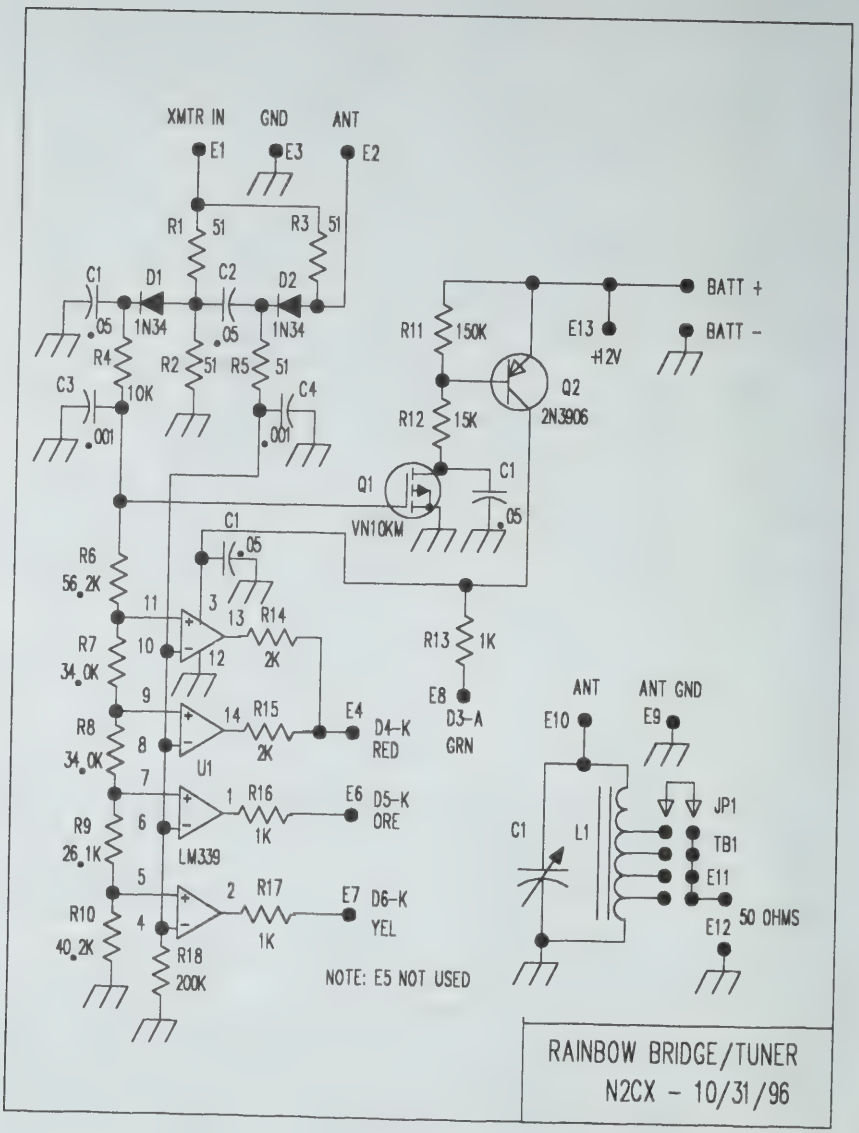
Q1, the MOSFET power switch is specially chosen to have a very low turn on voltage. I measured a sample of 8 devices and none of them took more than 1.5 volts dc gate voltage to turn on. In-circuit checks showed that none of them needed more than 150 mW of rf to activate the circuit. If you try a replacement, test it to be sure that it turns on at the power level you will use.

72/73, Joe Everhart, N2CX

Parts Kits for the Rainbow Tuner are available from the New Jersey QRP Club. The cost of the kit is $25, and includes the pcboard, board mounted parts, LEDs, sockets, but does not include the case or connectors. To order the kit send $25 plus

$3 shipping and handling for US addresses, $5 shipping and handling for non-US addresses. Make check or money order out to James Bennett, not NJ QRP Club. US Funds only. Send order to:

James Bennett, KA5DVS
309 Morrison Ave.
Hightstown, NJ
08520



Second Annual NorCal Dayton Building Contest

NorCal QRP Club is pleased to announce the second annual building contest in conjunction with the Dayton Hamvention. Builders may enter either or both of two projects; the "38 Special" or the "Rainbow Tuner" featured in this issue. True to NorCal tradition, the rules of this contest are simple: The project must work. Packaging is up to the builder, modifications are fine, we are interested in seeing your efforts. Chuck Adams, K5FO has agreed to be the judge again this year, and he will draft two assistants from the attendees to help. Prizes will be awarded to the first 3 places in each category for the projects judged to be the best by the judges.

Entrants may build their projects from homebrew parts, or they may purchase kits from NorCal for the "38 Special" transceiver and the New Jersey QRP Club for the Rainbow Tuner project. See page 7 for ordering information for the "38 Special" and page 11 for information on the Rainbow Tuner.

Judging for the contest will be Saturday night at 8:00 at the hospitality room sponsored by the QRP ARCI at the Day's Inn Dayton South during the Dayton Hamvention. Good luck and hope to see you there!

Homebrewing a Software Keyer

by Daniel Wee, 9V1ZV
daniel@pandora.lugs.org.sg
danwee@singnet.com.sg
daniel.wee@f516.n600.z6.fidonet.org

Introduction

Recently there has been some discussion about alternative keyers in the light of the demise of the venerated Curtiss 8044 keyer chip and its variations. From that discussion it appears that using small microcontrollers as keyers is becoming a fashionable trend, but also a powerful one that allows us to add features as we go along. However, it also came to light during that discussion that there is a shortage of source code available for those who would like to understand the workings of such a device. The lack of source code also means that those who would like to customize the code are unable to do so. This article is an attempt to shed some light in this area. Hopefully it is simple enough to be understood by non-programmers.

The Software and Hardware

The keyer software was originally written in C for the IBM PC compatible machines as part of my own logging software. It read the paddle status from the joystick port a-la the CT logger, and keyed either a serial port or a parallel port pin. The idea of having a keyer in software is particularly appealing because it made it possible to add all kinds of bells and whistles without making hardware changes. However, the initial concept was a little difficult to arrive at but once you got it, it is easy.

To put it simply, the software looks at the paddle status at fixed intervals, very frequently. It does this regardless of whatever else the machine is doing and in computer terms, we call this an interrupt driven routine. Essentially I re-programmed the PC timer to trigger off at much more than the 18 times a second it normally does. The frequency at which your program looks at the paddle determines:

A. The highest keying speed available

B. The resolution/steps of different speeds available

C. Reliability of the detection of paddling

This interrupt routine that looks at the paddle status is central to the keying program. Every time it checks to see if the paddle has changed status. If it has, then it proceeds to respond according to which paddle has changed. Since it looks at both paddles simultaneously, we can easily implement an iambic keying behavior.

The second part of this interrupt driven routine is to process current keying. The only difference between a dot and a dash and a space for that matter, as far as the software is concerned, is the length of the element and the keyed state (high or low). When a dash is started, the routine sets a countdown cycle to the length of that dash. Every time the routine is interrupted it reduces the count. Likewise a dot, and a space except that with a space the keyed state is reversed.

Also, after every element, there is a mandatory 'space' of the length of a dot before a second element is started. The computer buffers any paddle state changes during all this so that dot/dash memory is also implemented.

Changing the keying speed is a matter of changing the length of the elements, in fact the dot element in particular as everything else derives from this. The dash is three times (depending on weigthing) as long, and the space is the same length as the dot. Below is a C version of the routine that does the paddle checking and the keying:

The C Code:

```
unsigned int  keyspeed = (WPMMAX
+WPMLIM)/2;
int dittime;
int nodittime;

void interrupt timer_interrupt()
     {

/* This is to keep calling the old clock in-
terrupt at the correct rate
*/
secrate++;

/* Here is where we actually keep track of
the state of the paddles. 'kstate' is the cur-
rent paddle status while 'kdot' and 'kdash'
are previously read status for the dot and
dash respectively. This routine compares
if there has been any change in either
states.*/
kstate=inportb(JPORT)&48;
if (!kdot && !(kstate & 32)) kdot=2;
if (!kdash && !(kstate & 16)) kdash=2;

/* This checks for the mandatory spacing
and looks to see if the spacing timer is still
counting down. If it is, no element genera-
tion is allowed until the countdown is com-
pleted. It also checks if an element counter
is currently counting down is the space
counter is not in force. If so, it will also
decrement the element counter and upon
reaching zero, it will unkey the output.*/
if (nodittime) {
     if (dittime) {
          dittime—;
          if (!dittime) {
//             nosound();
               outportb
(keyport,keyoff);
                    }
               }
          if (!dittime) nodittime—;
          }

/* If there are no space or element counters
in action this means that the condition is
```

clear to generated a new element. If there has been no changes in the key state then this section is skipped. Otherwise it will generate an element depending on what has been read from the paddle status.

In order to implement the iambic mode, and dot/dash memory, we have to keep track of the different states that the keyer is in. kdot=2 means that a dot is pending but not started. kdot=1 means that a dot is underway. And kdot=0 means that no dot is present.

Remember that we will only reach here if all the counters have reached zero. So if the routines finds that a dot was previously underway, it will now reset the status to 0. Likewise for a dash. Also, if a dot was detected while a dash was previously underway, then we will insert a dot. Likewise for the otherway round.

If both paddles are pressed then it will alternate the elements.*/

```
else {
     if (kdot || kdash) {
          if (kdot==1) {
                    kdot=0;
          if (kdash==2) {
               dittime=keyspeed*3;
               nodittime=keyspeed;
               kdash=1;
               outportb(keyport,keyon);
//             sound(800);
                         }
                    }
               else if (kdash==1) {
                    kdash=0;
                    if (kdot==2) {
                    dittime=keyspeed;
                    nodittime=keyspeed;
                    kdot=1;
                    outportb
                    (keyport,keyon);
//                  sound(800);
                         }
                    }
     else if (kdot==2 && kdash!=1) {
          dittime=keyspeed;
```

```
            nodittime=keyspeed;
            kdot=1;
            outportb(keyport,keyon);
//          sound(800);
                        }
        else if (kdash==2 && kdot!=1) {
            dittime=keyspeed*3;
            nodittime=keyspeed;
            kdash=1;
            outportb(keyport,keyon);
//          sound(800);
                        }
                    }

/* This next section is irrelevant to us.
*/
    if (secrate%DIVRATE==0) {
        asm {
            cli;
            }
        (*oldvec2)();
        }
    outportb(0x20,0x20);
    }
```

Summary of C Code:

As you can see, although the code is rather long, the basic idea behind it is quite simple. It does require us to do some parallel processing in our head to keep track of what is going on though. However, the sections are quite definite and each block deals with a specific condition. This makes it easier to understand.

Of course, there are many ways that a keyer can be implemented and this is the one I chose because I found it intuitive. Being interrupt driven also allowed the other parts of my program to do what they liked without affecting the quality and timing of the keying. When I originally implemented this as a loop, the timing was unsatisfactory. The drawback of this method is that keying speeds are constrained, by the frequency at which this routine is called.

Microcontroller Version 87C51

When I was building a microcontroller keyer for my ARK-20, I chose the 87C51 because:

A. It was easily available
B. I had the programmer for it
C. I had the DataBook for it
D. The CMOS version took up less power
E. It had plenty of horsepower for my application
F. It had many available I/O pins (32 altogether)
G. It had internal EPROM and RAM
H. It required only an external crystal and 2 caps to get going

There are many more reasons but these are among the more critical ones for me. I suppose I could have used some of the PIC's which are much smaller but I needed the extra I/O pins for the other part of my application which the PIC did not have. I did have to add a small 8-pin serial ADC to read the speed control potentiometer and a transistor for keying, although this is not strictly necessary. You can do away with the pot altogether if you prefer to use up/down buttons for the speed control but since the unit has no non-volatile RAM, you will always power up at the default speed. With the pot, it will read the speed setting on the port.

Also, the code is probably portable to the many variations of the 8051 family MPU's and is quite small by most standards actually.

With the microcontroller, I ported most of the C code into 8051 assembly language and made some modifications to make use of the more flexible interrupt structure available in the microcontroller. The basic blocks are still the same, however. You have a routine that generates interrupts at a high frequency that calls the interrupt service routine. The ISR (interrupt service routine) does most of what the above C routine does. Below is the code:

8051 Assembly Code for Keyer:

```
$Title(Port Blinker)
$Date(03-29-96)
```

```
$MOD51

defcnt equ    1600
definv equ    65535-defcnt+1
dchi   equ    definv/256
dclo   equ    definv-dchi*256
WPMMAX equ    10
WPMLIM equ    63
addr   equ    3 ; start-bit, ch0 single ended
stretch equ   33

keyport equ   P2
keyer  bit    keyport.7
stcon  bit    keyport.6
pdotk  bit    keyport.5
pdshk  bit    keyport.4

kst    data   20h
kdotk  bit    kst.5
kdshk  bit    kst.4

adcclk bit    keyport.3
adcdo  bit    keyport.2
adcdi  bit    keyport.1
adccs  bit    keyport.0

keymask equ   0FEh
ksdelay equ   50
inispd equ    (WPMMAX+WPMLIM)/2
stk_ptr equ   5Fh
divsr equ     256/(WPMLIM-WPMMAX)

keyspd data   0Fh
noditt data   10h
ditt   data   11h
kdot   data   12h
kdash  data   13h
lastat data   14h
curstat data  15h
result data   16h

myflags data  21h
keytyp bit    myflags.0

       org    0000h
       ljmp   Start

       org    0003h
       reti

       org    000Bh
       clr    TR0        ; stop Timer 0
       ljmp   Timer_Int  ; service in-
terrupt

       org    0013h
       reti

       org    001Bh
       reti

       org    0023h
       reti

       org    100h
Start:
    mov   ie,#0    ; turn off all interrupts
    mov   sp,#stk_ptr  ; initialize stack

    mov   a,#07Fh  ; initialize all
ports, except keyer
    mov   keyport,a
    clr   keyer

    clr   a   ; initialize keyer variables
    mov   kdot,a
    mov   kdash,a
    mov   noditt,a
    mov   ditt,a
    mov   myflags,a
    mov   lastat,a
    mov   a,#inispd
    mov   keyspd,a

    mov   TMOD,#01h ; Timer 0 as
16-bit timer
    clr   TR0  ; stop Timer 0
    mov   TL0,#dclo    ; pre-load
16-bit count
    mov   TH0,#dchi
    setb  ET0      ; enable Timer
0 interrupt
    setb  TR0      ; restart Timer 0
    setb  EA   ; start all interrupts
    acall Ckstat
```

```
spdl1:
        clr     adcclk
        clr     adccs           ; chip-select,
start conversion
        mov     a,#addr
        mov     b,#3            ; number of bits
for address
loop1:                  ; send MUX address
        rrc     a
        jc      one
zero:
        clr     adcdi
        sjmp    cont
one:
        setb    adcdi
cont:
        acall   Pulse ; read out data (1-3)
        djnz    b,loop1
:       acall Pulse     ; sync clock pulse (4)
        mov     b,#8
loop2:
        acall   Pulse ; clock in ADC data
        clr     c
        jnb     adcdo,carry; read in ADC
                                data bit
        setb    c
carry:
        mov     a,result
        rlc     a ; shift in MSB first
        mov     result,a
        djnz    b,loop2
        setb    adccs ; de-select ADC,
                        EOC

        mov     a,result; scale and convert
                        data
        mov     b,#divsr
        div     ab
        add     a,#WPMMAX
        mov     keyspd,a ; update
                                keyspeed

        acall   Ckstat
        sjmp    spdl1

Pulse:
        setb    adcclk
        nop
        nop
        clr     adcclk
        ret

Ckstat:
        setb    stcon
        jnb     pdshk,tstk0; if either
                        paddle grounded
        jnb     pdotk,tstk0
        clr     ET0
        clr     TR0
        clr     stcon       ; setup for test
        jb      pdotk,tstk0
        setb    stcon
        setb    ET0
        setb    TR0
        mov     curstat,#0  ; nothing in
                                socket
        sjmp    tstk1
tstk0:
        setb    stcon
        setb    ET0
        setb    TR0
        mov     curstat,#1; something
                                plugged in
tstk1:
        mov     a,curstat
        cjne    a,lastat,chgtp1
        sjmp    chgtp2
chgtp1:
        mov     lastat,a ; update last plug
                        state
        jz      chgtp2
        clr     keytyp      ; assume paddle
        jb      pdshk,chgtp2 ; check if it
                        is straight key
        setb    keytyp
chgtp2:
        ret

Timer_Int:      ; Timer 0 ISR
        push    psw
        push    acc   ; save accumulator
        push    b     ; save divisor
        mov     TL0,#dclo ; reload 16-bit
                        count
        mov     TH0,#dchi
        setb    TR0 ; restart Timer 0
```

```
                        immediately
    mov     kst,keyport
;   mov     a,keyport
;   mov     kst,a
    jb      keytyp,strtk1; if straight
                          key
    jb      kdotk,til1    ; if (!kdotk
    mov     a,kdot        ; && !kdot)
    jnz     til1
    mov     kdot,#2       ; kdot = 2
til1:
    jb      kdshk,til2    ; if (!kdshk
    mov     a,kdash       ; && !kdash)
    jnz     til2
    mov     kdash,#2      ; kdash = 2
til2:
    mov     a,noditt
    jz      til3          ; if (noditt)
    mov     a,ditt
    jz      til4          ; if (ditt)
    dec     a             ; ditt—
    mov     ditt,a
    jnz     til4          ; if (!ditt)
            clr     keyer         ;
outport(keyport,keyoff)
til4:
    mov     a,ditt
    jnz     til5          ; if (!ditt)
    dec     noditt        ; noditt—
til5:
    sjmp    tilend
strtk1:                   ; straight keying
    jb      kdotk,strtk2
    setb    keyer
    sjmp    tilend
strtk2:
    clr     keyer
    sjmp    tilend
til3:
    mov     a,kdot
    orl     a,kdash
    jz      tilend  ; if (kdot || kdash)
    mov     b,#stretch;   percentage
                          stretch
    mov     a,kdot
    cjne    a,#1,til7     ; if (kdot == 1)
    clr     a
    mov     kdot,a        ; kdot = 0
    mov     a,kdash
    cjne    a,#2,tilend ; if (kdash == 2)
til11:
    mov     a,keyspd
    mov     noditt,a ; noditt = keyspeed
    add     a,keyspd
    add     a,keyspd
    mov     ditt,a ; ditt = 3 * keyspeed
    div     ab
    add     a,ditt
    mov     ditt,a
    mov     kdash,#1      ; kdash = 1
    setb    keyer;outport
                    (keyport,keyon)
    sjmp    tilend
til7:
    mov     a,kdash
    cjne    a,#1,til8     ; if (kdash = 1)
    clr     a
    mov     kdash,a       ; kdash = 0
    mov     a,kdot
    cjne    a,#2,tilend ; if (kdot == 2)
til10:
    mov     a,keyspd
    mov     noditt,a ; noditt = keyspeed
    mov     ditt,a        ; ditt = keyspeed
    div     ab
    add     a,ditt
    mov     ditt,a
    mov     kdot,#1
    setb    keyer ; outport
                    (keyport,keyon)
    sjmp    tilend
til8:
    mov     a,kdot
    cjne    a,#2,til9 ; if ((kdot == 2)
    mov     a,kdash
    cjne    a,#1,til10 ; && (kdash !=
                          1))
til9:
    mov     a,kdash
    cjne    a,#2,tilend ; if ((kdash ==
                          2)
    mov     a,kdot
    cjne    a,#1,til11 ; && (kdot != 1)
tilend:
    pop     b     ; restore divisor
    pop     acc   ; restore accumulator
```

```
pop     psw
reti            ; end of Timer_Int ISR

end
```

Summary of 8051 Assembly Code

As you can see, this closely corresponds to the C code although there are some subtle changes. There is also included the routine which reads the National Semiconductor ADC0832 analog to digital convertor for the keying speed.

The code begins with some definitions and declarations. This is followed by initialization code at the beginning which initializes the variables and sets up the timers, interrupts and ADC. Start: begins the main loop. Ckstat: is the routine that checks the paddle inputs to see if the key type has changed. It uses a stereo socket for the paddle input and if a straight key is pulgged in, it will automatically change modes to straight keying mode. This does not work too reliably however, so I eventually checked the key type only at power up.

Timer_Int: is the routine that is called by the interrupts and is the complement of the C routine provided above. It differs in a few ways however. I added a 'stretch' variable to adjust the weigthing a little so as to make the feel a little better.

To fully understand this program, one would have to know quite a bit about the 8051 architecture and assembly language which I don't really want to get into here. There are many good books on this though and you can probably find it is any good bookstore since the 8051 is just about the most popular microcontroller ever.

This code is a fairly early incarnation but is straightforward to understand. I eventually incorporated other features which reduced the frequency at which is checked the ADC and other stuff so reduce QRM from the ADC mostly. I also added some protection resistors to the paddle inputs ince I accidentally damaged one MPU via the paddle port. A later version of this source code is included with the schematics at the end of this article.

Construction:

Below is a copy of what I originally wrote to describe the construction of the keyer plus some correspondence about it:

Whew! Finally got that keyer working. What keyer? Oh, the one I've been working on. Lately I've been thinking of adding a keyer to my ARK-20 which only accepts a straight key. It would be easy for me to whip up a very small Curtiss 8044 based keyer to put it inside BUT that would mean that I'll be stuck with a paddle and not be able to use the straight key. I toyed with the idea of adding a toggle switch to select the paddle/key but the idea did not seem terribly elegant. Finally, after much debating I decided to build a special keyer that would automatically distinguish the type of key in use. As there wasn't much space in the ARK-20, I had to look at a very compact solution. Hence my keyer project was born.

The keyer uses an 87C51 microcontroller and measures 6.5 by 2.8 cm (not inches!). Most of the parts sits in the space under the CPU in the space inside the CPU's 40-pin socket. The parts count is fairly low:

*1	87C51(Philips) 40-pin MCU	
*1	ADC0832 (NS) 8-pin A/D convertor	
1	78L05	5V regulator
*1	2N2222	driver transistor
5	0.1uF	mylar caps
*2	10uF/16V	electrolytic caps
1	4.7k/0.25W	resistor
*1	8.2k/0.25W	resistor
*1	11.0592 MHz	crystal
*2	22pF	ceramic caps
1	68mH	RFC
*1	40-pin	turn-pin socket
*1	50k variable resistor (off-board)	

It seems a lot, but remember, this includes the 5V regulated power supply, along with excessive decoupling and RF blocking parts. If you already have a 5V line and prefer to use that, then your parts count would be even lower. Likewise, if you remove excess decoupling capacitors, the parts count goes down even further. The '*' marks the essential parts. The rest are just there to play safe. I didn't want RF from the keyer getting into the RX front-end since they're mounted in the same enclosure. The 50k variable is not critical, since it only acts as a voltage-divider. I suppose even a 100k value would work here.

When assembled, 3-wires come off for the variable resistor. 2-wires for the keying. 4-wires for the paddle/key. The configuration is as follows:-

To original key port connections:

1 Keyer
2 Gnd

To speed control pot:

1 +5V to variable resistor (1)
2 Variable resistor wiper (2) speed control
3 Gnd to variable resistor (3)

To stereo socket with break sense connection:

1 Dot
2 Dash
3 Sense
4 Gnd

To power supply:

1 +7 to +12V power supply
2 Gnd

A lot of redundancy in the ground connections could be avoided in the actual installation but I just wanted to be consistent. There are also some optional connections.

To serial port for computer control (optional):

1 RX-serial
2 TX-serial
3 Gnd

To ARK-20 thumbwheel for frequency reading (future):

1-16 Data pins

To any analog voltage from 0-5V (optional):

1 Extra ADC channel
2 Gnd

I intend to evetually implement computer control for my ARK-20 using the format for the CI-V of ICOM radios. I haven't gotten round to this yet but the hardware is already in place.

After many rounds of program/debug/UV-erase/program/etc. I finally got the keyer to work correctly. Learned a lot about the 8051 along the way. This looks like it could be a very useful part.

I also picked up some Hitachi LCD displays really cheap at about US$3.00 each. I have a neat way of adding this to the ARK-20 in mind.

On the whole, I am very pleased with the result because of its small form-factor, and the possible new things I can implement in software without changing too much hardware. Furthermore, I believe this is the only auto-keytype-sensing keyer I have seen. You can pre-set the keying speed limits in the software without having to change any parts. I haven't actually installed it into my ARK-20 just yet but have tested it out quite thoroughly.

The 87C51 MCU has about 4K of EPROM space, of which I used just a few hundred bytes, leaving plenty of room for creativity. There are lots of unused (as yet) port lines. You could also try out the 89C1051 or the 89C2051 MCU's by Atmel which are code compatible and much smaller. I did not use these parts because I cannot find them locally, nor did I have a programmer for them. The total cost of parts (built on wire-wrap style breadboard) is about US$20 or so.

This is because I am using a UV part for the MCU. If you use an OTP part, it will be much cheaper. The single most expensive part is the 87C51 which cost me some

US$14.00 or so. In the US this might be cheaper.

Below is the code I used in case any one is interested. You will notice that it is pretty short but it is all in assembly language. I'd be pleased to entertain any questions about the circuit.

A memory keyer is a definite possibility but you'd have to add one of those serial EEPROMs from Microchip so that the memory is retained even after power off. The additional hardware is trivial, most of the work will be in the software. You'd also have to put in a few more buttons but the rest is implemented in software, and running serials is okay provided you can think of a scheme to let the keyer know you'd like to have that inserted.

As a Memory Keyer:

This is a distinct possibility with the main obstacle being the inclusion of some form of non-volatile memory to the circuit. The easiest way to do this, as far as I can tell, is to use one of those 8-pin serial EEPROMs such as those made by MicroChip. This requires only a few lines to interface. The rest of it includes adding a few push buttons to the ports. The coding is a simple exercise in programming and should fit nicely into the rest of the 4k of unsed EPROM space. The reason I did not already do this was because my application was primarily as a keyer for the ARK-20 and in that case (literally) there was simply no space for any additional pushbuttons. Besides, I wanted to save the available port pins (a little over 16 unused as yet) for another purpose. There is nothing to stop anyone else from using these pins for their own purposes however.

As a Frequency Counter

Another easily implemented feature. The 87C51 has a externally triggerable counter. Someone responded that he had actually implemented just such a frequency counter. This could be something a lot of people might want for their little rigs. The coding for this is probably quite simple in view of the fact that the MCU hardware directly supports such functions.

As a Morse Code Decoder:

This can easily be done, with or without the help of an external PLL such as the NE567. The decoded output could be transmitted using the serial port on the MCU (which will require a MAX232 to convert to regular RS-232 levels if interfaced using normal RS-232 serial ports). Alternatively, you could pick up an LCD display (I got a few at US$3.00 each) and interface this to the MCU with 7 port pins. These are intelligent displays so nothing more is needed except perhaps a negative voltage source (may not be needed).

I have built quite a few morse decoders in my time and found their performance lacking usually and the operation of such devices finicky. As a result I have little faith in automatic morse decoders. What they say about the best decoder being the one between our ears is true.

However, the ease of interfacing an LCD to the MCU does raise new possibilities. If I were not so constricted for space, I would have added an LCD to display code speed, frequency, and perhaps some other info.

Using up/down speed control buttons instead of a variable resistor:

In fact I have another version of code which controlled the speed of the keyer using two push buttons (or one centered DPST spring loaded toggle switch). The reasons I went with the variable resistor version were:

1. I did not have space for the pushbutton(s). Actually I do but it was not the right size. The ARK comes with a hole pre-drilled for a variable resistor. I suppose if you really wanted to, you could put in a 2 position toggle switch with a centred normally open contact. This would avoid the need of the ADC.
2. The more pressing reason was that with the pushbutton scheme I had to set the button to my preferred speed upon every

power-up since my circuit did not include any non-volatile RAM. However, if we did choose this scheme, the space vacated by the ADC would nicely accept an 8-pin serial EEPROM. This was not availble on hand.

Avoiding the need of the 2N2222 driver transistor:

It was pointed out that in fact, if I used PORT-0 on the 87C51 to drive the keying circuit directly, I can avoid the need of a driver transistor. This is correct, due to the special fact that PORT-0 is an open collector output without pull-ups. In my case, I needed all the 8-pins of PORT-0 for some other purpose, and so I did not use PORT-0. The additional space used by the 2N2222 did not result in any increased board size.

Using pulse-width measurement to measure resistance instead of ADC:

This was a possibility I explored too. The idea is that I could use a port pin to charge up a known capacitance and to discharge it through the variable resistor at another pin. Upon charging up the capacitor, it presents a logic-1 to a third pin, which then starts a timer and the discharge circuit. After a period of time proportional to the resistance value, the charge of the capacitor would have dropped below the logic-1 threshold and the MCU now has a count proportional to the resistance of the variable reistor. This scheme requires only one capacitor and possibly one other resistor for calibration purposes and uses only 3 or 4 port pins. I was not sure of the stability of the measurements yielded using this method because the resolution would be proportional to the conversion time. To achieve good resolution would require substantially longer periods. Further more, since the keyer was interrupt driven, it may interfere with the timing of the pulse width, or vice-versa. Nevertheless, this is one possibility worth exploring further and the same technique could come in useful elsewhere.

The Spare ADC Channel:

The ADC0832 used actually has 2-input channels so you could easily use the other channel for your own purposes, such as another variable resistor, etc. without the need of adding any additional hardware.

Conclusion:

As you can see, the use of an MCU does greatly increase the number of features that can be included in a project for a relatively small increase in parts count. With the prices of today's MCU's it would be a sin for the QRP community to continue to ignore the hidden potential here. Many newer projects could actually employ serially driven PLL VFO's such as those made by Motorola, easily with an MCU in the rig. LCD displays also add to the professionalism of the final product.

Something that I realized was that there is a lot of flexibility in designing circuits with MCUs. For example, in my prototype board, the ADC was not under the MCU. In my final circuit, I decided to put it under the MCU. Doing this put the interface pins of the ADC directly next to some port pins. So instead of pulling wires from the original pin assignments to the new ADC position, I simply changed the source code to reflect the new pin assignments. The result, 4 wires less under the board. This kind of configurability is very handy when space is a constraint.

The 8051 and its relatives come in a variety of different flavors and sizes. Some come with a built in ADC/DAC, extra counters, higher speeds, more EPROM space, more internal RAM, etc. Some come in 20-pin packages, making them really small, and most require the least of external support circuitry to make them work. Furthermore, there are other parts such as the PIC 16Cxx parts by MicroChip, or the Basic Stamp by Parallax, all of which represent very easy to use controller parts which could greatly enhanced the features of any small homebrew rig, and possibly reduce the cost. The keyer I built cost about

as much as a good Curtiss based keyer, plus I get to make it behave exactly the way I want it to behave and the option of adding other features to it.

The drawback of using microcontrollers in QRP rigs is primarily the lack of programming tools required to work with these parts. One would need to invest in a device programmer (which may vary from part to part) and perhaps a UV eraser, as the very minimum. The software can usually be obtained for free from the net. These would include cross-assemblers, simulators, etc. This problem can partly be circumvented by using parts which can download programs from the PC serial port, and thus not needing any specialized programming equipment.

The Parallax BASIC STAMP is one such part. You write your program in BASIC on the PC, compile it and down load it to the very tiny part. THE BSC-1 and 2 are so small you'd have to see it to believe it. They support serial port functions, accept BASIC programs, and has plenty of pins, and the BSC-1 comes in an SIP form so the foot print is exceptionally small. It is a little costly however.

A second problem involved with using microcontroller parts is the generation of RFI. Since there is almost always an oscillator involved, and the digital lines are mostly carrying square waves, you can and should expect lots of harmonics to be generated. This can usually be dealt with by careful PCB layout and screening, proper decoupling and careful part selection. One can also select a crystal frequency that does not interfere with the rig in question. Remember, most of the modern commercial rigs contain MCUs anyway, and they seem to have worked around this problem by and large.

Finally, remember that the most valuable asset of any homebrewer is his (or her) creativity. MCUs gave me a chance to be more creative, it could do the same for you.

Note:

1. The ADC0832 used is the 8-pin DIP variety. There is also a 14-pin SOP type which I have not tried. The pin-outs here are for the 8-pin DIP variety.
2. The value of the 50k speed control potentiometer is not critical. This simply works as a voltage divider and any suitable value will do. It should not be too extreme a value (1K or less, or 100K or more).
3. Not shown in the diagram above are decoupling capacitors for both the IC's. These are 0.01uF capacitors soldered across the power supply pins of the IC's to ground. Decoupling capacitors were also added on most of the 87C51 port lines to reduce RFI. Experiment.
4. The port pins can be reassigned to suit your board layout. Look at the source code and change the assignments in the code to suit your requirements.
5. The "Socket N.C." goes to the little tag on the stereo socket which is normally closed. Be sure to get a stereo socket with this if you wish to use the auto key detect feature.
6. As noted in the article, the keying transistor (any suitable NPN, 2N2222 or 2N3904) can be eliminated altogether if one of the Port 0 pins are used. This is not shown here.
7. The crystal marked Y1 is an 11.0592 MHz crystal. The value is probably not critical if you do not plan on using the serial port. In that case, a 12 MHz or 10 MHz crystal should do fine. You might have to scale it a little differently in the code. WPMMAX and WPMLIM may need to be tweaked to give you the speed control range you want on the potentiometer.
8. The 22pF capacitors around the crystal are tiny ceramic ones. I used NP0 but this is not necessary. The rest of the capacitors were small monolithic caps. I chose these for their size more than anything else.
9. Also not shown here is the 78L05 regu-

lator I used to convert the 12V from the rig into the 5V required by this circuit. Again, liberal use of decoupling and smoothing capacitors were used. In fact, I had a big valued inductor in the Vcc line from the 12V to the regulator to prevent QRM from flowing back into the rig.

10. My construction was on a matrix board which individual pads. Interconnections were made with wire-wrapped wires which I soldered point-to-point. This allowed me to cram most of the parts into the space in the 40-pin IC socket. This is the gap found under the 87C51. I had a few resistors on the solder side of the board as well but these were mostly because they were added as an after thought.

11. Many of the values in this circuit are non-critical. Resistors especially, and the keying circuit can be changed and/or improved.

12. The ADC032 can also be replaced with an ADC08032. Both are National Semiconductor parts.

13. The ADC can be eliminated altogether and replaced with up/down buttons for speed control. Some changes in the code must be done for this to work.

14. The 510 ohm resistors were added later to protect the I/O port pins of the MPU from destruction. Greater values are probably better as long as the paddle is being read properly.

15. The reset electrolytic, 2.2 uF, is not critical. It should be rated more than 6V at least. I picked the smallest one I could find. The value is also not critical. A 1 uF value will probably work just as well.

16. Observe that the keying circuit consists of an open collector transistor. Make sure that you don't try to key high voltage loops with this configuration. As it is, this is suitable for most QRP rigs. If you plan to key valved rigs, you'll have to incorporate a more robust keying section. You might want to add a reverse biased diode across the keyed output and ground to prevent damage from reverse polarity circuits. This is probably not necessary for QRP rigs but discretion should be used.

17. Since this circuit has a high frequency oscillator in it, be sure to keep all wires as short as possible. Use screening where possible. I made a screening cover out of some pieces of double-sided PCB cut and soldered together.

Below is the current source code for this keyer with lower-RFI enhancements. The CI-V computer control code is included but can be removed if not desired. This is simply two calls in the main loop, both of which are commented. This code was compiled with a public domain 8051 compiler called the Metalink Assembler and is available from many FTP sites.

Program: ML-ASM51.ZIP
Description: MetaLink's 8051 family macro assembler
Location:
ftp.pppl.gov : /pub/8051/signetics-bbs
ftp.funet.fi : /pub/microprocs/MCS-51/signetics-bbs

```
$Title(Port Blinker)
$Date(03-29-96)
$MOD51

        defcnt  equ     1600
        definv  equ     65535-defcnt+1
        dchi    equ     definv/256
        dclo    equ     definv-dchi*256
        WPMMAX  equ     10
        WPMLIM  equ     63
        addr    equ     3; start-bit, ch0 single
                            ended
        stretch equ     33

        ARK equ         027h; Machine
                                address

        lobcd equ       P0
        hibcd equ       P1
        rxen bit        P3.0
        txen bit        P3.1
```

```
kst     data    20h
kdotk   bit     kst.5
kdshk   bit     kst.4

keyport equ     P2
keyer   bit     keyport.7
stcon   bit     keyport.6
pdotk   bit     keyport.5
pdshk   bit     keyport.4

adcclk  bit     keyport.3
adcdo   bit     keyport.2
adcdi   bit     keyport.1
adccs   bit     keyport.0

keymask equ     0FEh
ksdelay equ     50
inispd equ (WPMMAX+WPMLIM)/2
stk_ptr equ     57h
divsr   equ     256/(WPMLIM-
                WPMMAX)

keyspd  data    0Fh
noditt  data    10h
ditt    data    11h
kdot    data    12h
kdash   data    13h
lastat  data    14h
curstat data    15h
result  data    16h
oldlo   equ     17h
oldhi   equ     18h
cmdlvl  equ     19h
tmpsbuf equ     1Ah
curcmd  equ     1Bh
latency equ     1Ch

myflags data    21h
keytyp  bit     myflags.0

org             0000h
ljmp            Start

org             0003h
reti

org             000Bh
clr             TR0  ; stop Timer 0
ljmp            Timer_Int ; service
                        interrupt

org             0013h
reti

org             001Bh
reti

org             0023h
reti

org             0050h
Start:
mov             IE,#0; turn off all
                        interrupts
mov             IP,#0; normalize
                        priorities
mov             SP,#stk_ptr; initialize
                        stack
mov             a,#07Fh;initialize all
                ports, except keyer
mov             keyport,a
clr             keyer

mov             a,#0FFh; initialize
                radio ports for input
mov             lobcd,a
mov             hibcd,a
setb            rxen ; enable serial
                        receive
setb            txen ; enable serial
                        transmit

clr             a  ; initialize keyer
                        variables
mov             kdot,a
mov             kdash,a
mov             noditt,a
mov             ditt,a
mov             myflags,a
mov             lastat,a
mov             cmdlvl,a; serial
                        command level
mov             a,#inispd
mov             keyspd,a
```

```
        mov     a,#10; initial latency
        mov     latency,a

        mov     TMOD,#21h ; Timer
                0 as 16-bit timer;
                Timer 1 as 8-bit auto-
                reload

        clr     TR0 ; stop Timer 0
        clr     TR1; stop Timer 1

setb    PT0     ; prioritize keyer
setb    IT1     ; edge triggering
mov     SCON,#50h   ; Mode 1,
                8-data bits
mov     TH1,#0E8h; 1200 baud

mov     TL0,#dclo; pre-load 16-bit
                count
mov     TH0,#dchi
setb    ET0 ; enable Timer 0
                interrupt
setb    TR0 ; restart Timer 0
setb    TR1 ; restart Timer 1
setb    EA  ; start all interrupts
acall   Ckstat; check key type

mov     a,lobcd
mov     oldlo,a
mov     a,hibcd
mov     oldhi,a

lcall   sndstr
db  13,10,`Blink-CIV 1.2 01/08/96
Daniel Wee (c) 9V1ZV ARK-
20',13,10,0FFh

acall   send_md
acall   send_fr

spdl1:
    jb      keytyp,spdl2
    acall   Ckadc; check keyer speed
                setting
spdl2:
;   acall   Ckstat; check if key type
                has changed
mov     b,latency

blpu1:
    acall   Ckfreq; check if frequency
                has changed
    acall   Ckcmd ; check if C-IV
                command received
    djnz    b,blpu1
    sjmp    spdl1

Mydata:
    db  0FEh,0FEh,ARK,0E0h,0,0FDh

Pulse:
    setb    adcclk
    nop
    nop
    nop
    nop
    clr     adcclk
    ret

Ckadc:
    clr     adcclk
    clr     adccs ; chip-select, start
                conversion
    mov     a,#addr
    mov     b,#3  ; number of bits for
                address
loop1:  ; send MUX address
    rrc     a
    jc      one
zero:
    clr     adcdi
    sjmp    cont
one:
    setb    adcdi
cont:
    acall   Pulse  ; read out data (1-3)
    djnz    b,loop1
:   acall   Pulse ; sync clock pulse (4)
    mov     b,#8
loop2:
    acall   Pulse ; clock in ADC data
    clr     c
    jnb     adcdo,carry; read in ADC
                data bit
    setb    c
carry:
    mov     a,result
```

```
        rlc     a  ; shift in MSB first
        mov     result,a
        djnz    b,loop2
        setb    adccs; de-select ADC,
                        EOC

        mov     a,result ; scale and convert
                         data
        mov     b,#divsr
        div     ab
        add     a,#WPMMAX
        cjne    a,keyspd,ckad1
        mov a,latency
        cjne a,#50,ckad3
        sjmpckad2
ckad3:
        inc  latency
        sjmpckad2
ckad1:
        mov latency,#5
        mov  keyspd,a     ; update keyspeed
ckad2:
        ret

Ckstat:
        setb  stcon
        jnb   pdshk,tstk0   ; if either paddle
grounded
        jnb   pdotk,tstk0
        clr   ET0
        clr   TR0
        clr   stcon         ; setup for test
        jb    pdotk,tstk0a
        setb  stcon
        setb  ET0
        setb  TR0
        mov     curstat,#0      ; nothing in
socket
        sjmp  tstk1
tstk0a:
        setb  stcon
        setb  ET0
        setb  TR0
tstk0:
        mov     curstat,#1      ; something
plugged in
tstk1:
        mov   a,curstat
        cjne  a,lastat,chgtp1
        sjmp  chgtp2
chgtp1:
        mov   lastat,a ; update last plug state
        jz    chgtp2
        clr   keytyp  ; assume paddle
        jb    pdshk,chgtp2   ; check if it is
straight key
        setb  keytyp
chgtp2:
        ret

Ckfreq:
        mov a,lobcd
        cjne a,oldlo,ckfl1
        mov a,hibcd
        cjne a,oldhi,ckfl1
        sjmpckfl3
ckfl1:
        mov a,lobcd
        mov oldlo,a
        mov a,hibcd
        mov oldhi,a
        acall send_fr
ckfl3:
        ret

Timer_Int:  ; Timer 0 ISR
        push  psw
        push  acc  ; save accumulator
        push  b  ; save divisor
        mov     TL0,#dclo  ; reload 16-bit
count
        mov   TH0,#dchi
        setb  TR0           ; restart Timer 0
immediately
        mov   kst,keyport
;       mov   a,keyport
;       mov   kst,a
        jb    keytyp,strtk1  ; if straight key
        jb    kdotk,til1     ; if (!kdotk
        mov     a,kdot       ; && !kdot)
        jnz   til1
        mov     kdot,#2      ; kdot = 2
til1:
        jb    kdshk,til2     ; if (!kdshk
        mov     a,kdash      ; && !kdash)
        jnz   til2
```

```
        mov     kdash,#2        ; kdash = 2
til2:
        mov     a,noditt
        jz      til3            ; if (noditt)
        mov     a,ditt
        jz      til4            ; if (ditt)
        dec     a               ; ditt--
        mov     ditt,a
        jnz     til4            ; if (!ditt)
                clr     keyer                   ;
outport(keyport,keyoff)
til4:
        mov     a,ditt
        jnz     til5            ; if (!ditt)
        dec     noditt          ; noditt--
til5:
        sjmp    tilend
strtk1:                         ; straight keying
        jb      kdotk,strtk2
        setb    keyer
        sjmp    tilend
strtk2:
        clr     keyer
        sjmp    tilend
til3:
        mov     a,kdot
        orl     a,kdash
        jz      tilend          ; if (kdot || kdash)
          mov   b,#stretch      ; percentage
stretch
        mov     a,kdot
        cjne    a,#1,til7       ; if (kdot == 1)
        clr     a
        mov     kdot,a          ; kdot = 0
        mov     a,kdash
        cjne    a,#2,tilend     ; if (kdash == 2)
til11:
        mov     a,keyspd
       mov      noditt,a        ; noditt = keyspeed
        add     a,keyspd
        add     a,keyspd
       mov      ditt,a          ; ditt = 3 * keyspeed
        div     ab
        add     a,ditt
        mov     ditt,a
        mov     kdash,#1        ; kdash = 1
                setb    keyer                   ;
outport(keyport,keyon)
        sjmp    tilend
til7:
        mov     a,kdash
        cjne    a,#1,til8       ; if (kdash = 1)
        clr     a
        mov     kdash,a         ; kdash = 0
        mov     a,kdot
        cjne    a,#2,tilend     ; if (kdot == 2)
til10:
        mov     a,keyspd
       mov      noditt,a        ; noditt = keyspeed
        mov     ditt,a          ; ditt = keyspeed
        div     ab
        add     a,ditt
        mov     ditt,a
        mov     kdot,#1
                setb    keyer                   ;
outport(keyport,keyon)
        sjmp    tilend
til8:
        mov     a,kdot
        cjne    a,#2,til9       ; if ((kdot == 2)
        mov     a,kdash
        cjne    a,#1,til10      ; && (kdash !=
1))
til9:
        mov     a,kdash
       cjne     a,#2,tilend     ; if ((kdash == 2)
        mov     a,kdot
        cjne    a,#1,til11      ; && (kdot != 1)
tilend:
        pop     b               ; restore divisor
       pop      acc             ; restore accumulator
        pop     psw
        reti                    ; end of Timer_Int ISR

sndchr:                         ; transmit a se-
rial character
        clr     TI
        mov SBUF,a
sncl1:
        jnb     TI,sncl1
        ret

sndstr:                         ; send an FF ter-
minated string
        pop  dph
        pop  dpl
```

```
sndl1:
      clr   a
      movc      a,@a+dptr
      cjne a,#0FFh,sndl2
      sjmpsndl3
sndl2:
      acall sndchr
      inc   dptr
      sjmpsndl1
sndl3:
      clr   RI
      mov a,#1h
      jmp       @a+dptr

send_md:
      lcall     sndstr
      db
0FEh,0FEh,0E0h,ARK,04h,03h,0FDh,0FFh
      ret

send_fr:
      lcall     sndstr
      db
0FEh,0FEh,0E0h,ARK,03h,00h,0FFh
      mov       a,lobcd
      acall     sndchr
      mov       a,hibcd
      anl       a,#1Fh
      acall     sndchr
      lcall     sndstr
      db        014h,0,0FDh,0FFh
      ret

Ckcmd:        ; Check for C-IV
                commands
      mov       dptr,#Mydata
      jnb       RI,ckcml1
      mov       a,SBUF
      clr       RI
      mov       tmpsbuf,a; store away
                received char
      mov       a,cmdlvl
      cjne      a,#4,ckcml3; is this
                command position?
      mov       a,tmpsbuf
      mov       curcmd,a
      sjmp      ckcml5
ckcml3:
      movc      a,@a+dptr; compare string
                position
      cjne      a,tmpsbuf,ckcml2
      cjne      a,#0FDh,ckcml5
      mov       a,curcmd
      cjne      a,#4,ckcml4; mode
                request?
      acall     send_md
      sjmp      ckcml2
ckcml4:
      cjne      a,#3,ckcml2; frequency
                request?
      acall     send_fr
      sjmp      ckcml2
ckcml5:
      inc       cmdlvl; increase command
                level
      sjmp      ckcml1
ckcml2:
      clr       a
      mov       cmdlvl,a
      mov       curcmd,a
ckcml1:
      ret

      end
```

Disclaimer:

This whole project is presented as is. I have done my best to make sure that everything is right but it should be noted that there is a very experimental nature to projects such as these. I can say that it works very well for me but many more enhancements to hardware and software can still be made. As with all projects, think of what you are doing. That should help prevent some major disasters. Good luck.
73 de 9V1ZV Daniel

Arizona ScQRPion Mini-Mag Paddles

by Floyd Smithberg, NQ7X

With all the paddles to choose from today at very nominal cost (?) why bother to even consider building one? Unless, perhaps like me, you get your kicks more from building than operating... or, also like me, you're somewhat 'financially deprived.' this little paddle, 1.5" x 1.5" x 3" without base) is the most recent of many attempts at making a small, cheap, easily constructed model from parts available at any hardware store and with simple hand tools e.g. electric drill, files, hacksaw, etc.

Earlier versions were: an electronic tube keyer with built-in paddle in 1952, spdt momentary key switches with modifications to limit trhow and provide positive stops, various mechanical bugs, etc. No doubt there will be others in due time.

The action on this one is controlled by a small magnet (3/16" x 3/16" x 7/8") embedded in the wooden base below and in contact with the 4-40 steel flathead machine screw. The two paddle arms are also steel and attracted by the vertically mounted screw.. they also provide the center ground contact. The fixed adjustment contacts are 2-56 round head machine screws to provide fine adjustment, with their tips filed to a point. The arm pivots were cut to approximate length from 1/8" steel rod then 'machined' to a point and length using a hand drill and file, and finally polished with emory cloth.

The top and bottom brass plates were cut to size with a hack saw from 1/16" x 1" brass stock and after laying out the location for the holes were clamped together and drilled through both plates to get exact alignment. Then to get precise alignment of the paddle arms and pivot, locate them on a piece of wood and use small brads to keep in place while soldering. I considered using silver plated contacts but after using these paddles with two versions of the KC-1, my CMOS III, TS850, and FT736 with no problems, it appears not to be necessary. However, in more corrosive environments than Phoenix, you might consider other contact materials.

Use the drawings and the pictures to duplicate this design. You should be able to do it with less than $10 in materials plus some time that you will enjoy. When you finish, you will have a set of paddles that are sturdy, tough, and work well. It works great for backpacking, mobile or home use. Please let me know if you come up with mods or improvements.

Construction:

Building the Arizona Scorpion MiniMag Paddles does not take any special equipment, but it does take care attention to laying out the pieces and making sure that the holes are drilled accurately. The following procedures will help you build a set of paddles that you can be proud of.

First measure and cut the brass stock for the bottom and top plates. These are made from 1/16" x 1" brass stock that is available from any hobby shop that sells model airplanes. Look for the K&S Brass display.

Now lay out the drilling holes on the brass. This is done by careful measurement from one end of the brass base plate. The best method to drill in metal is to use a process called step drilling. In this case, start all holes using a 1/16" bit, then finish the hole with the proper sized bit. You will find that your holes are much more accurate using this method. Use a drill press if at all possible, as the holes need to be vertical. Also, please use a center punch to mark the start of the drill, as otherwise your drill bit will tend to run on you. The holes for A & B are marked on Top Plate 1 and it is then clamped to the bottom plate and the holes are drilled at the same time. This assures perfect alignment. Next, mark the holes for Top Plate 2, and clamp it in position over the bottom plate. Again, drill

with the proper size drill using a drill press if at all possible. Finally, mark the rest of the holes that are in the bottom plate and drill.

Turn the bottom plate over and counter sink the holes where required. Tap the threads for the hole at C. Lay these parts aside and find the steel 1/8" rod. Cut two pieces exactly 1 and 5/16" long. Put them in an electric drill held in a vise. Use a file to turn a point on each end of the rod, trying to keep them the same length as much as possible. Next, cut the contact arm to length. The covers for the slots in a computer are excellent for this. Lay the contact arm on a piece of wood, and hold it using small brads. Position the pivot on the arm, make sure that it is exactly perpindicular to the arm, and use brads to hold it in place. Solder the pivot to the arm, making sure that the positon of the pivot matches the dimensions in Figure 1.

The next piece is the base. This can be any material that you choose. Wood, plastic, steel, brass, aluminum, all will work well here. You will need to cut a cavity out of the base for the magnet. The easiest way to do this is to get a router bit and use it in your drill press. Take it slow, and it will work fine. Be sure to use guides to keep control of the work piece. Or you can drill a series of holes to the proper depth and use a wood chisel to finish the job. Don't worry how it looks, as it will be covered and won't show.

Finally, prepare the Nylon spacer. Tap both ends of the spacer for threads to fit the screws that will hold it on. Then lay the spacer on its side and drill a hole through the side. Tap for 2-56 threads.

Now we are ready to assemble the paddles. Assemble as shown in the drawing in Fig. 2. Use the screws at position B to adjust the bearing tension on the paddle pivots. Place the magnet underneath the screw at D, and adjust the play on the contacts on the nylon posts. Wire the paddles, with the dits connected to position B in Figure 2 and the dahs connected to position A. The ground connection goes to position C. That is all there is to it. Good luck!

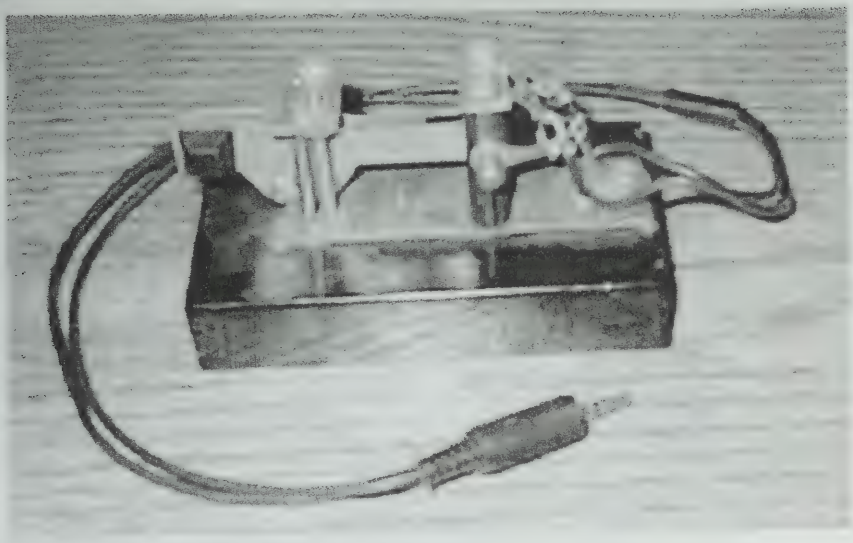
Parts List:

Bottom Plate Brass	2 1/2 x 1 x 1/16"
Top Plate 1	5/8 x 1 x 1/16" Brass
Top Plate 2	3/8 x 1 x 1/16" Brass
Magnet	3/16 x 3/16 x 7/8"
Contact Arms (2)	2.5 x .5 x .045 Steel
Pivots (2)	1/8 x 1 5/16" Cold Rolled Steel
FH Brass Machine Screws (2) 6-32	1.5" x
RH Brass Machine Screws (2) x 8-32	1/4"
FH Steel Machine Screw (1) 4-40	1" x
RH Brass Machine Screw (1) x 6-32	1/4"
FH Steel Machine Screw (2) x 8-32	1/4"
FH Brass Machine Screw (1) 6-32	1" x

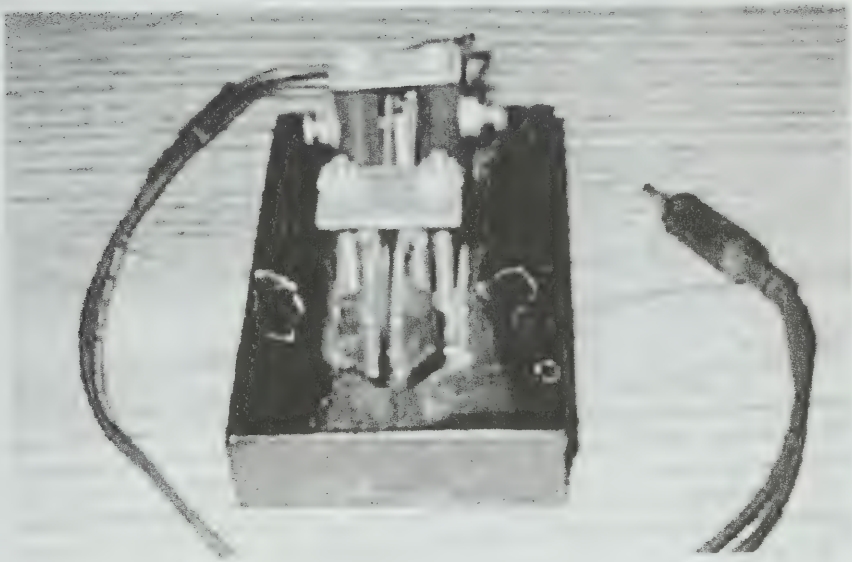
(3) Solder Lugs, (2) 6-32 Brass Nuts, Base Material

Note: The paddle may be made with the top as one solid piece, but if you choose this option, please note that you will have to make sure that the two front posts which connect to the paddle cable must be insulated from the top piece. The easiest way to do this is to use nylon mounting screws and nylon washers if you use metal for the front posts. Of course, if you use nylon standoffs, there is no problem.

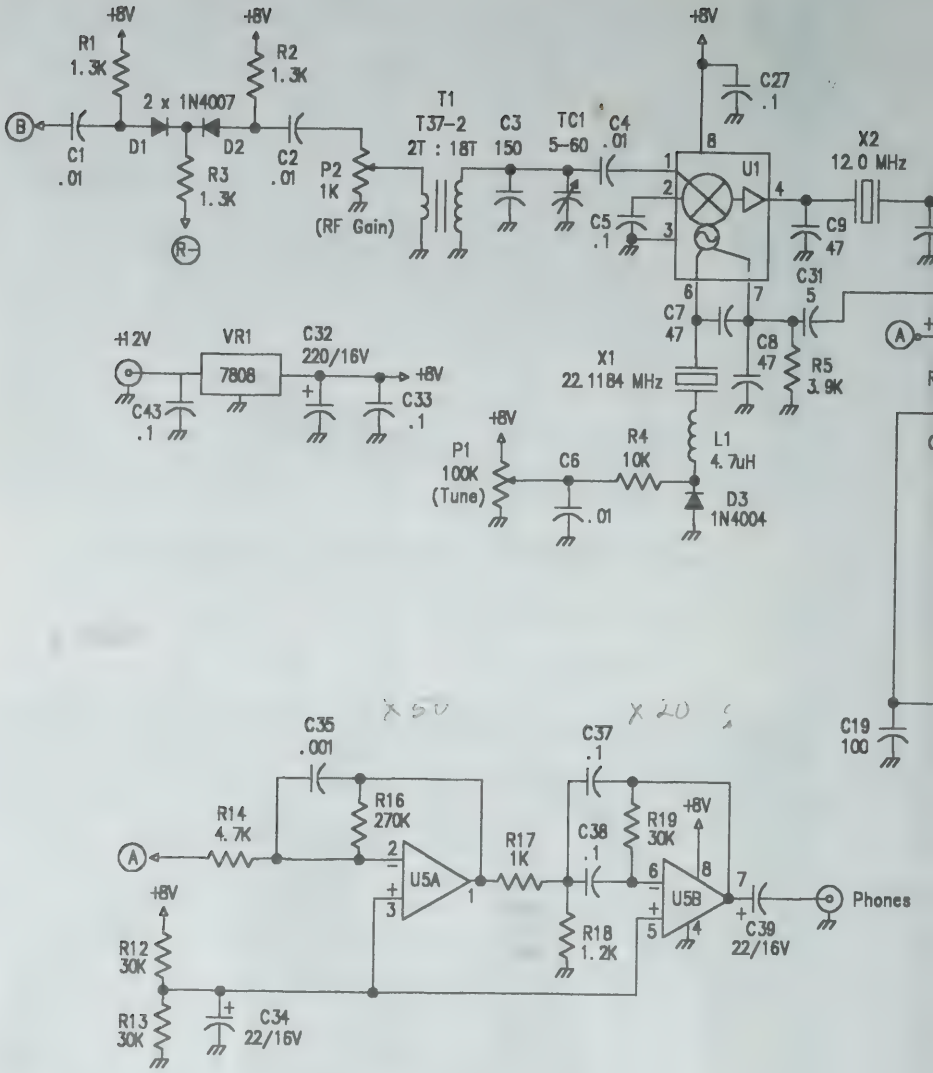
To use a one piece top, you will have to mount the top and then use a cut and try method for the bearings for the paddle arms.



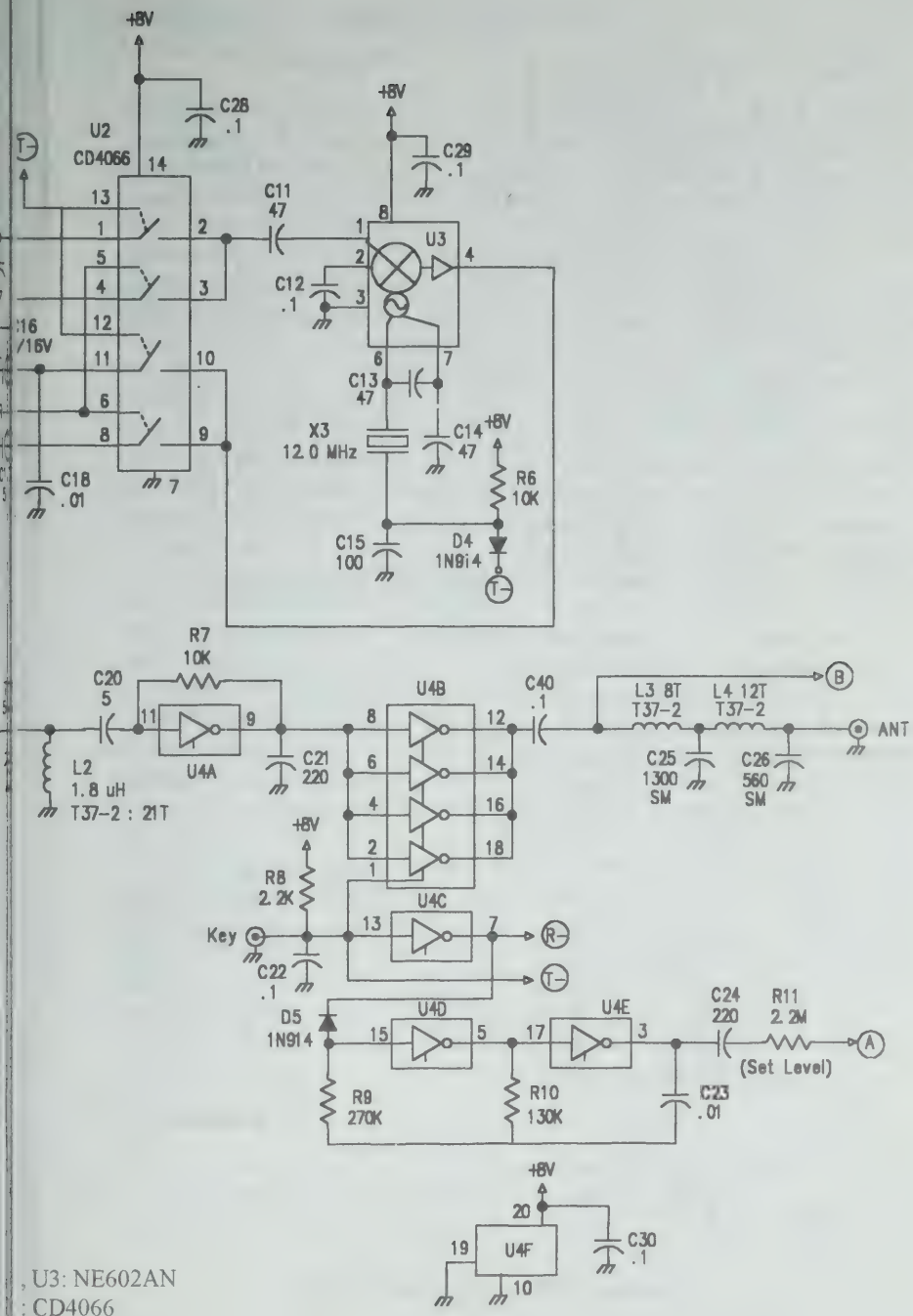
Side View, NQ7X Version Arizona Scorpion Mini-Mag Paddles



End View, NQ7X version of Arizona Scorpion Mini-Mag Paddles

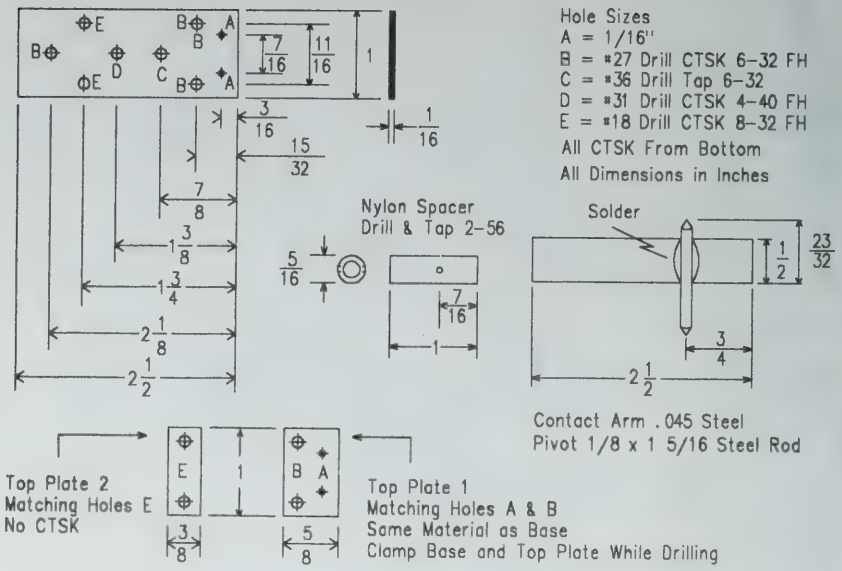


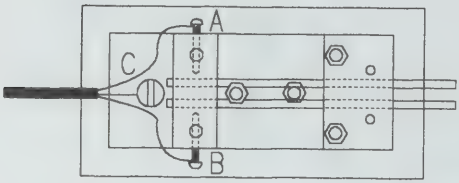
"38 Special"
30 Meter CW Superhet Transceiver
designed by Ori Mizrahi-Shalom, AC6AN
Winner of the 1997 Dayton Building Contest Design Contest
Rev. 1.0, Nov. 1, 1996



, U3: NE602AN
: CD4066
: 74HC240
: NE5532
cimal Caps are in uF, All others in pF

Arizona Scorpion Paddles
NQ7X

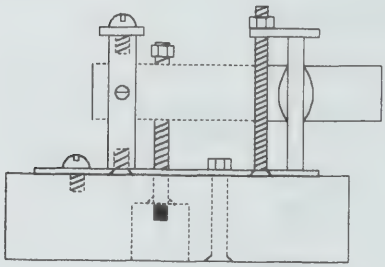

Hole Sizes
A = 1/16"
B = #27 Drill CTSK 6-32 FH
C = #36 Drill Tap 6-32
D = #31 Drill CTSK 4-40 FH
E = #18 Drill CTSK 8-32 FH
All CTSK From Bottom
All Dimensions in Inches
Nylon Spacer
Drill & Tap 2-56
Solder
Contact Arm .045 Steel
Pivot 1/8 x 1 5/16 Steel Rod
Top Plate 2
Matching Holes E
No CTSK
Top Plate 1
Matching Holes A & B
Same Material as Base
Clamp Base and Top Plate While Drilling



Top View

Refer to Fig. 1 for dimensions. Not all parts shown in all views for clarity.

Dark area in cutout represents magnet placement.



Side View

End View

Arizona ScQRPion Mini-Mag Paddles

A Mighty Fine Tuner

by John Spoonhower, KC2DU

Here's a quick and simple construction project for those of you that need an easy-to-build long-wire tuner. I was inspired by the post of Craig, WB3GCK, to QRP-L some time ago. I built the tuner he mentioned in a dead MFJ noise bridge that I picked up at the Rochester, NY hamfest this past spring. Hence the name....a Mighty Fine Tuner. The noise bridge board was in a pretty bad way, but the tuning capacitor and connectors were in good shape. So the board went into the junk box, and the other components and case were pressed into service for this little tuner. The tuner is ideal for backpacking/camping as it's quite small.

The tuner schematic is shown in Figure 1. With the switch in the "down" position, the tuner becomes an "L" circuit with the antenna wired at the capacitor/inductor junction. This configuration is good for matching wire antennas that are multiples of a half wavelength. With the switch "up", the inductor and capacitor are in series, and better suited to match quarter wavelength long wires. Nice and simple....

My version of this tuner is shown in Figure 2. I have to confess that I really don't like to do a lot of mechanical work in my construction projects. With the old MFJ box, even "mechanically-challenged" hams (like myself) can produce decent looking results without a lot of effort. Of course, any small enclosure (2 x 3-1/4 x 4 in) can be used here. Note that the controls for the inductor and capacitor are on opposite sides of the box. I was interested in minimizing drilling of the case and, hey, we 're camping anyway.....

The only drilling necessary in my case was to open up the diameter of the hole for the tuning capacitor a little bit. The capacitor is mounted to a small piece of plexiglass (perspex for you G-guys!) epoxied to the inside of the case. The shaft and mounting hardware for the capacitor cannot contact the case in this circuit. The capacitor has a value of roughly 300 pF in my circuit; Craig, WB3GCK, suggests 200 pF minimum. The inductor is hand wound #22 wire onto a 35 mm plastic film can. There are a total of 40 turns producing an inductance of 35 microhenries. The coil is tapped 8 times; a Radio Shack rotary switch selects the different tap points. I suggest you connect a counterpoise wire when using this tuner in the wild. I have a tie point mounted on the outside of the case to clip on such a wire with an alligator clip.

This is a really minimal-pain project that really useful results. It's about the easiest way to get started in homebrewing that I can imagine. If you're lucky enough to have a dead MFJ, well then you can REALLY minimize the pain!

72, John, KC2DU

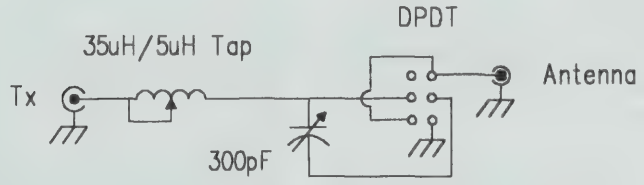
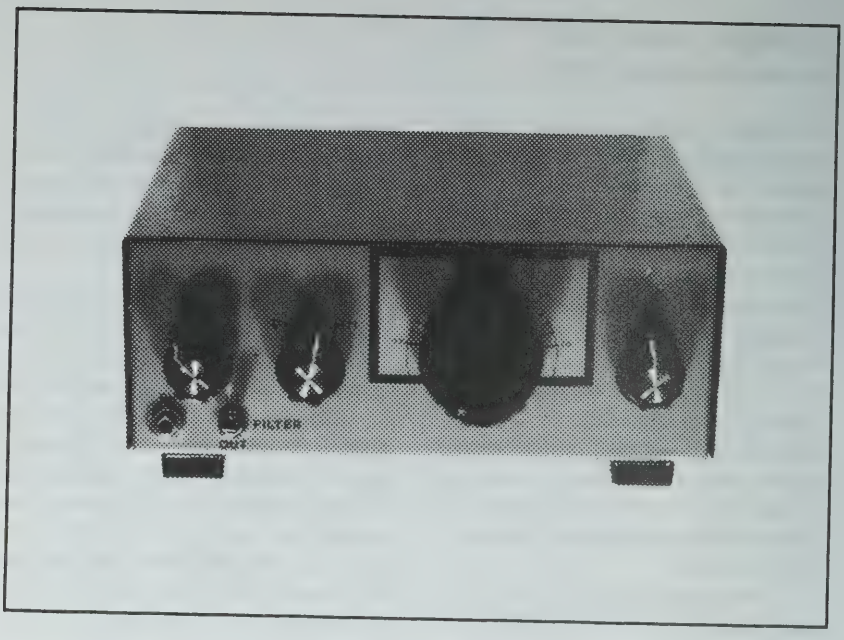


Figure 1



Fix the Thump in your NW80-20 Series Rigs

by Paul Harden, NA5N
pharden@nrao.edu

When I was at Dayton, I was approached by several people asking if I knew the cause of a "thump" on key-up on the EmTech NW series rigs. This was curious, as my NW40 does not exhibit any funny noises during keying. Finally, Preston Douglas WJ2V sent me his NW20 and NW80 ... which indeed are champion "thumpers." But they did allow me to finally find the problem(s) - which are applicable to other rigs as well.

1. LEAKY DIODE D1 (1N4148). This is the keying diode that biases the FET audio mute switch. It is located next to the VFO tuning capacitor, behind the audio pot on the board. On two of the rigs I checked, this diode was leaky and not allowing the FET mute switch to go into full turn-off on key-down. REPLACE WITH A 1N4001. Roy at EmTech now supplies a 1N4001 for this diode in his kits. I just snipped D1 off the top of the board and soldered the 1N4001 on the remaining leads to avoid disassembly of the board from the enclosure. MORAL: Ensure your diodes are diod'ing. On a DVM, the forward resistance measured about 60K vs. almost nothing.

While this fix cured a couple of the NW thumps, it did not cure Preston's thump.

2. RF PICKUP ON AUDIO FILTER OUTPUT. There are two flavors of the NW rigs with enclosures. One where the CW filter on/off switch is on the LEFT side of the front panel, and the other where the switch is on the RIGHT side of the front panel. Where the CW filter switch is on the RIGHT side, the wiring from the CW filter board (left side of rig) runs across the board to the right side, right over the RF stuff. This wiring acts as an antenna and picks up a bunch of RF on key down. The induced RF is applied to the FET mute switch (audio input) and fed back to the CW filter op amps. On key down, this RF

charges up the filter output cap and the cap across the FET mute switch. On key-up, it takes 800mS for these caps to discharge sufficient to again pass audio. When the discharge occurs, there's a big appearance (thump) of audio.

PLACE A .1uF CAP ON THE CW FILTER AUDIO OUT TO GROUND. To avoid disassembling the rig, I mounted the .1uF cap on the CW filter on/off switch ... wrapped one lead around the switch shaft for ground, the other lead to the switch terminal that receives the audio from the CW filter board. This completely eliminates the thump; a .05uF was not quite enough. Your mileage may differ. A .05uF will cause less audio attenuatio if sufficient in your case ... although there's plenty of audio to spare.

The alternative is to add the cap on the CW filter board from the OUT terminal to ground. Works equally as well, but requires some disassembly.

Another approach on your NW rig may be to just route the wiring from the CW filter board to the switch by tucking it along the inside of the front panel so its not dangling in mid-air over the RF drivers. There was no thumping on those rigs with the CW filter on/off switch on the left, or just an inch from the filter board. MORAL: Be conscious of routing wiring in and around RF power stages. Bypass low-frequency or DC wiring if needed to keep the RF out.

I could not see this "thump" on the o-scope, making it even more frustrating. For your interest ... what I finally did is hooked up a square wave generator with a TTL (0v/5v) output to the key jack to key the rig. I slowed down the square wave generator until the thump appeared, which was around 12Hz. I triggered the scope off the square wave and used the 2nd trace to go probing around, mostly central to the FET mute switch. This is where I first saw the capacitor discharging 800mS later. Previously, I was probing around while tapping the key, and 800mS was just off the sweep, and because my "tapping" was asynchronous, was not able to see it. Triggering to a stable square wave, allows one to see what is happening many milliseconds (in this case, almost a second) later. At 20wpm sending the character "V", the thump would occur only after the "dah." However, the scope would trigger on the first dit and the problem occured long after the sweep finished. Slowing the sweep down to the "seconds" range, the discharge pulse was just too narrow to see. Thus is took a square wave generator triggering on the falling edge to see it. I offer this explanation due to the several emails I've received how people can hear the thump, but not see it on a scope.

I might add, both of Preston's NW rigs were well built, good solder joints throughout, tight toroids, etc. It was just a serendipitous routing of switch wiring that caused RF to be induced during transmit, and a leaky diode.

NW owners with the thumping ... check diode D1 for proper switching action and that the audio input to the FET mute switch is free of RF by bypassing as described above. This does seem to be specific to those using the CW audio fiter with the filter on/off switch located on the right side of the front panel. Once rid of the thump, you will also find the variable IF filter and the CW audio filter work much smoother together for tighter/quieter bandwidths! If you have any questions, feel free to email me about it, or the designer, Roy W6EMT at roygregson@aol.com.

Finally, though I misplaced his calls, I would like to thank Bob Barry for also discovering the leaky diode part of the problem with a rig or two on his end. While I had found the problem, I wrote it off as a once-in-a-lifetime failure. Through email, it became evident Roy had received a bum batch of diodes. The power of QRP-L and internet strikes again!

GL, Paul NA5N

Huff and Puff for the Epiphyte

by Ed Burke, KI7KW
28 Del Prado
Lake Oswego, OR 97035

About a year ago, I started getting interested in SSB QRP. I quickly discovered that there is a fundamental difference between CW, which had been my previous sole focus, and single sideband. Specifically, I found I needed to watch my frequency much more carefully on SSB; a few hundred Hz. driftg, which would have gone unnoticed in a fifteen minute QSO on CW, drew vigorous complaints on phone.

So, I started getting interested in miniature frequency counters with 100 Hz. resolution, and so on. And frequency readout devices are great, but it still seemed a shame to be forced to make frequency corrections manually. After all, if the box knows enough about the current tranceiver frequency to display it, it could darn well take care of correcting the error! But no such luck, and I was considering modifying one of my conters to close the loop.

I also thought about the fact that one really did not need the whole frequency display to stabilize it. Arguably, the last digit would be enough. If you could freeze the last digit to a plus or minus one Hz. range then the job would be done, and the remaining digits would take care of themselves.

I subscribe to QEX, the ARRL experimenters journal, and I noticed an article in the February issue titled "Frequency Stabilization of L-C Oscillatgors" written by Klaas Spaargaren, PA0KSB. After reading the article, I decided that the techniques presented had a good chance of working successfully, and I stopped my own efforts at inventing a frequency stabilizer. My technical specialty is control systems,though, so the problem had a firm grip on me.

I participate in a QRP SSB net on 75 meters at 3760 kHz., and mentioned the article to Derry Spittle, VE7QK and Willie Murray, VE7YY. They told me that the article described the old "Huff and Puff" circuit, and that the basic idea had been around for ages. In fact Derry sent me a copy of an old article from an RSGB journal which described an earlier version of the circuit. So now you understand the source of the weird title for this piece.

At its core, the PA0KSB circuit is based on comparing the VFO to a reference frequency (conveniently generated from a crystal oscillator) using a high speed digital mixer. When the two frequencies are locked, nothing happens, but when the VFO drifts, it generates error pulses, which are summed up in an integrator circuit. The output of the integrator is used to vary the bias on a conventional varicap diode (such as an MVAM108) which corrects the VFO frequency to tune out the drift.

The correction is deliberately slow; it is intended to null out long-term thermal drift and will not control rapid transients. But with this circuit installed and operating in my Epiphyte 2, I have been able to routinely measure VFO stability of 5 Hz per hour. It ultimately depends on the stability of the crystal oscillator which is used as the reference.

Technically, the Huff and Puff is a frequency-lock-circuit with an "integral" characteristic. Integral control loops have the advantage that they drive the error reliably to zero, but htey have lousy dynamics, which is another reason why the response is deliberately glacial. Actually the slow correction is an advantage. When the transceiver is manually retuned to a different frequency, the change easily overrides the Huff & Puff's authority, so that I don't even notice that it is there.

The frequency correction response

looks a little like a saw tooth pattern (see figure 1). Using the example values in the QEX article results in a lock point every 11 Hz (marked "E"); that is, if the VFO is driven out of lock for any reason, it jumps to the next stable equilibrium point 11 Hz away. But only one half of that range is usable. If you assume a VFO which increases its frequency for increasing control voltage, then only the portions labeled "stable" are usuable.

As to practical details, I built in into an external mini-box (2"x2"x3") with a very short cable (about 3" long) and a four-pin connector to the Epiphyte case. It draws about 35 mA, so making it removable for low current battery operation seems desirable. The second version was built on a 2.5" x 2.5" piece of perf-board which resides on the inside of the lid of the transceiver. Surprisingly, there seems to be no serious interference between the Huff & Puff and the Epiphyte transceiver.

The design which was described in the QEX article used both 5 volt and 12 volt supplies but the Epiphyte 2 uses only 5 vdc. So I made some changes to allow the circuit to run on 5 volts only, and those changes are shown in the sketches below. Let me summarize what I modified from the design published in the QEX article:

* The PA0KSB design uses an old RCA CMOS operational amplifier, the CA3140. For 5 volt supply I used one of Texas Instruments new "rail to rail" CMOS op amps, the TLC2272.

* The non-inverting pin of the op amp is biased to 2.5 Vdc using a pair of resistors (instead of 5 Vdc).

* I changed the input coupling circuit to one that I stole from Derry Spittle because I wanted more sensitivity and less load on the VFO.

* Transistor leakage is a big concern with the original design. The 2N3904 and 2N3906 transistors can be expected to leak significantly in the off state, and the leakage is unlikely to be matched. In the original version of the circuit, leakage has no place to go but into the tntegrator inputs, so I have added two low value resistors which return the current to a stiff 2.5 Volt level which has been made from an unused op-amp section.

* For the coupling back to the varicap circuit, the QEX article recommends a sensitivity of 1 kHz/V. I found best results with a sensitivity of 2 kHz/V, which is realized with a 200K resistor from the output of the Huff and Puff to the cathode of the varicap diode in the Epiphyte. I also tried an additional varicap diode for the Huff & Puff, and that works fine too.

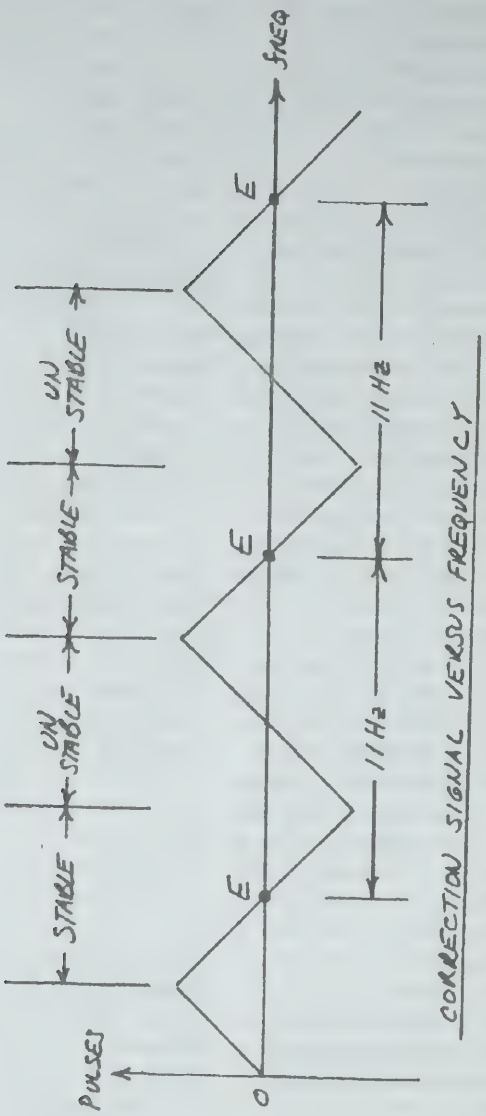
* The QEX article shows an example circuit with an overtone crystal oscillator at 48 MHz, but the usuable lock range (considering a VFO at 4.2 MHz) is only 5.5 Hz. the Epiphyte VFO tends to "rumble" a little, bouncing about 2 Hz up and down around its target frequency. So 5.5 Hz is too narrow a range. If you use an oscillator frequency of 35 MHz, that results in a lock range of more than 7 Hz, which works much better. By the way, consider using one of the minature DIP package crystal oscillators (see the Digi-Key catalog) in place of the 10 or so separate parts shown in the drawing; overtone oscillators can be touchy.

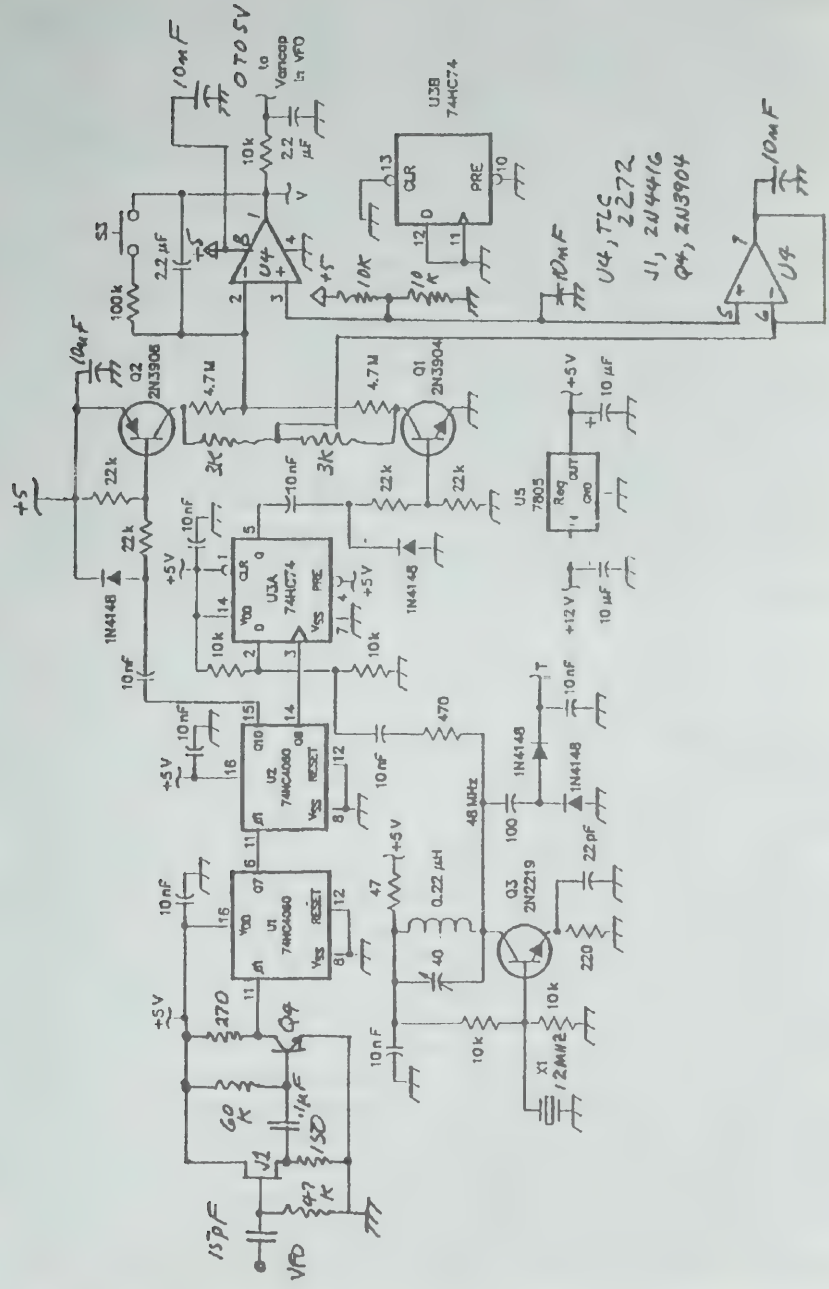
* Consider building the integrator portion of the circuit, which is very sensitive to stray currents at the nano-Amp level, using "ugly" or deadbug techniques, to avoid circuit-board leakage.

* Finally, be careful to use a VFO which is frequency stable over receive and transmit conditions. The VFO should have its own regulator, and even that may not be enough. In my Epiphyte, the 13.7 Volt input was pulled down to 13.5 Volts on transmit, and that change was enough to shift the VFO by 12 Hz and drive it out of lock. The cure is simple; use an additional regulator at 8 Volts, and have the 8 Volts feed the 5 Volt regulator (in cascade), then the VFO is rock solid when the little rig is

keyed.

If you would like the stability of your QRP rig to rival that of the typical 100 watt "appliance" with its crystal-controlled synthesizer chain, the Huff & Puff circuit is a good bet. Mine now works perfectly. Enjoy, Ed Burke, KI7KW




U4, TLC 2272
J1, 2N4416
Q4, 2N3904
+5
+12 V
VFO
S3
Q1 2N3904
Q2 2N3906
Q3 2N2219
Q4
U1 74HC4060
U2 74HC4060
U3A 74HC74
U3B 74HC74
U4
U5 7805
U6
X1
48 MHz
0.22 µH
1N4148
1N4148
1N4148
1N4148
10 µF
10 µF
10 nF
10 nF
10 n

Cascade Steel-Case Crystal Mod

by Dave Meacham, W6EMD
206 Frances Lane
Redwood City, CA 94062-2733
ddm@datatamers.com

This mod applies ONLY to the last 50 kits sold. They contain different crystals than those in the first 200 kits. This situation was beyond Doug's control. The modification improves the performance of the crystal filter.

The first 200 kits sold by NorCal had aluminum cases. The second batch (50 kits) had nickel-plated steel cases with the "JK" label. This mod is for the kits with crystals having the "JK" label.

The mod is based on computer simulation using Wes Hayward's GPLA program. I have had excellent results using this technique in the past. No one has built a filter using this mod, although at least one VE7 may be doing so soon. The mod consists of replacing all but one of the capacitors in the crystal filter with new ones having different values.

Predicted-performance comparisons follow:

Stock Values:
Bandwidth (-6dB): 4.1kHz
Return Loss: Deep valley at 28dB, non-symmetrical.
Passband: Rounded top.
Modified Values:
Bandwidth: (-6dB): 3.0kHz
Return Loss: 18dB, symmetrical.
Passband: Flat top.
The capacitor changes follow:
Modified Values:
Change C78 and C80 to 68pF.
Change C77 and C81 to 91pF.
Change C82 and C83 to 130pF.
(C79 stays at 56pF)

I recommend using mica capacitors if possible, otherwise NP0 or C0G ceramic types. 72, Dave, W6EMD

A Piezo-Electric Key

by Glen Leinweber,VE3DNL
110 Marlowe Dr.,
HAMILTON, On Canada L9C 2H9
leinwebe@mcmail.mcmaster.ca

Asks the devil's advocate, "Why complicate something as simple as an on/off switch with a mess of electronics?". And, "Don't all keys with no mechanical motion have a poor feel?". Ok, that's the downside. The upside is ruggedness, small size and almost no weight. One of these keys could be taped down to the outside case of your rig. With appropriate insulating coating, it would work under water. d\ the simple electronics described later use next to no power. Doesn't that sound like an almost ideal key for a backpacking rig?

Ceramic piezo-electric benders are usually used as an audio alarm in smoke detectors, quartz clocks, games, beepers. But the piezo effect works in reverse too - a mechanical stress generates a voltage output. Like a crystal, a thin wafer of piezo ceramic has both sides coated with a conducting surface. Commercial units are available from 0.5 to 2 inch diameter for $0.50 to $3.00. Most come with one side bonded to a thin brass plate, slightly larger in diameter than the ceramic wafer. Overall thickness is 0.02 inch or less. One electrode is connected to the brass disc, the other can be soldered to the coating on the other surface.

A ceramic piezo element looks elec-

trically like a capacitor at the very low frequencies that we are interested in - from D.C. up to about 20 Hz. Mechanical stress impresses a voltage across its two electrodes; positive when bent one way, negative when bent the other. Voltage magnitude is directly proportional to bending stress. Even light finger pressure can generate about a volt (depends on mounting) but there's no perceptible motion at any keying weight.

The brittle ceramic transducer should be mounted to a stiff, smooth surface. Fixing it down with solder is unwise: the ceramic will bend from thermal gradients, and possibly key the rig by itself. To counteract piezo voltages caused by temperature effects, a large resistance bleed was added in parallel. Secure the piezo down with celophane or electrical tape. But protect the coated surface - it may come off with the tape.

A high impedance amplifier connected across the electrodes is necessary to isolate a low impedance load, since load current would quickly discharge the stress-induced voltage. With piezo capacitance of about 0.01uf, an amplifier input impedance of much greater than ten MEGohm is needed at low keying speeds. The bleed resistor must be large too: 65 MEGohms allowed about three seconds keydown, plenty for very slow code.

The most simple interface would be a single enhancement-mode N- channel MOSfet like a BS170, VN10K, or IRF510 as in figure 1a. The zener diode protects the gate from overvoltage, since the MOSfet can withstand only +/-20 volts on its gate. The output at the MOSfet drain should be able to key most rigs - it seems to key a Norcal 40A cleanly. Keying may be too soft for some rigs - an amplifier with higher voltage gain may be required. Modern CMOS op-amps having input currents much less than 1 picoamp make ideal buffers, and can drive the logic gate inputs of a keyer, or a MOSfet keying transistor. Connect the transducer so that keying force gives a positive voltage on the gate. Otherwise, your key will work backwards.

Figure 1b shows a more exotic key. U1 is a CMOS op-amp, used as a comparator. Its output is either up at the supply voltage, or down at ground, depending on whether pin 3 voltage is above or below that of pin 2. Pot R2 adjusts the keying weight: a higher voltage requires more piezo stress to switch the op-amp output. U1 is a low-power device, requiring only 16uA from its supply.

Only one piezo element is needed for a keyer: "dot" force will generate a voltage of one polarity, "dash" force the opposite polarity. Figure 2 shows how two op-amps (used as comparators) could generate ouput voltages to drive the keyer dot and dash inputs. The piezo element is biased at half the supply voltage. R3 and R4 independently adjust dot and dash keying weights. To get enough keying weight range, this circuit may require a larger supply voltage: the rig's 12 volt supply should do. Some paddle users only apply keying pressure around the outside edge. That's not advisable with these piezo elements, since force near the middle gives highest voltage. Paddles could be added that would both re-distribute edge pressure to the center, and protect the transducer too.

Iambic keying needs two piezo elements which can be mounted on either side of a very stiff paddle. There is a danger that dot or dash stresses could bleed through to the opposing transducer. Two separate paddles with a transducer mounted on each would be safer. Each transducer would use the circuit of figure 1a or 1b.

What does it feel like? I doubt that anyone would prefer its feel over any other key. Feeling that contact "clunk" when a key closes is an important tactile aspect that is missing. Perhaps some will find novel mounting methods for mobile or portable use.

From ELECTROSONIC catalog (Willowdale Ont):

"Projects Unlimited" Piezo-ceramic benders:

Cat #	outside dia.	piezo dia.	overall thickness
KBI-1541	15mm	9mm	.23mm $.80 Cdn
KBI-2048	20mm	15mm	.33mm $.70 Cdn
KBI-2064	20mm	15mm	.43mm $.50 Cdn
KBI-2744	27mm	20mm	.53mm $.45 Cdn

Outside dimension is the diameter of the brass backing plate. Overall thickness includes the brass backing.

72, Glen Leinweber

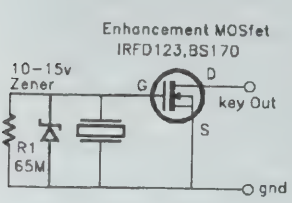


Figure 1(a)

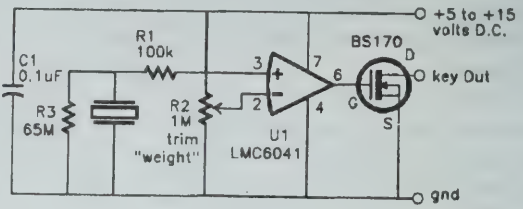


Figure 1(b)

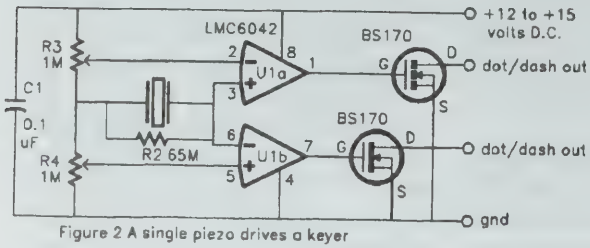
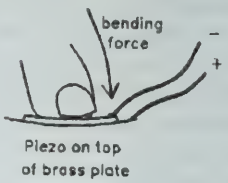
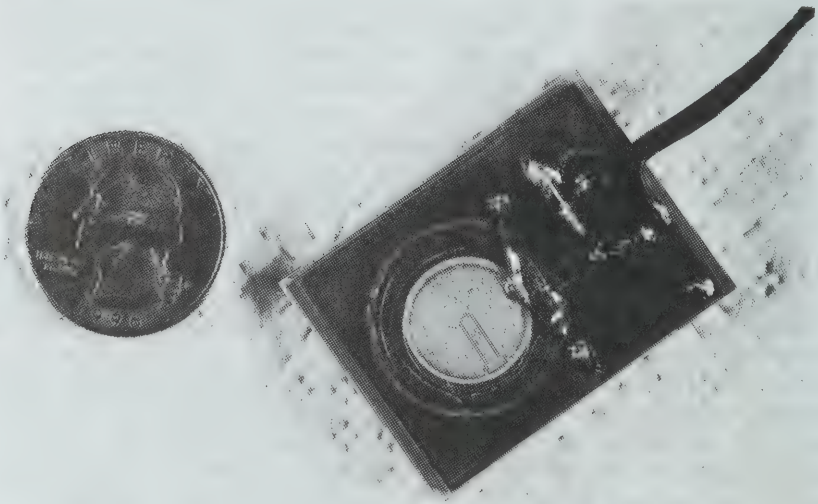


Figure 2 A single piezo drives a keyer



Piezo Schematic



SIERRA POWER OUTPUT - REVISITED

By Bob Follett, WA7FCU
2861 Estates Dr.
Park City, UT 84060
bfollett@ditell.com

The Sierra, designed by Wayne Burdick - N6KR, Kitted by NorCal, and now by Wilderness radio, is up to revision C, and has been the subject of many modifications in the QRP journals and on QRP-L. I constructed mine this summer in anticipation of the wonderful KC-2 "Swiss Army Knife" Keyer, frequency counter, watt meter and S-meter. Mine was assembled using the now-current instruction manual including recommended modifications, which will be covered later.

After testing the few band modules that I had assembled, I became concerned about the amount of RF output my WM-1 was recording. This lead me on a search for past Sierra modifications in the journals, e-mails with Wayne, Dave Meacham - W6EMD, and constant sharing of experiences with Preston Douglas,WJ2V and Bob Gobrick, VO1DRB. The manual's specifications state that power output (13.8V supply) is 2 to 3 watts on 1.8-14Mhz, and 1.5 - 2 watts above. However, Wayne, in a post to QRP-L, stated that his, essentially stock, Sierra puts out 3.5W on 160 & 80M, 3.0W on 40M, 2.3W on 20 though 15M, and 2W on 10M.

There are many reasons one may start the power output chase, and in my case, it was Wayne's post. I wanted my radio to work at least as well as his! Before going further however, it is important to state Wayne's design philosophy related to power output: "To precisely control drive level into each mixer takes many more components, raising the cost above the level where it is a viable kit. I opted to keep the cost down and accept compromise performance, especially since the rig is intended to be "low power" to be compatible with backpacking." Enough said.

There are two major classes of power seekers who may be interested in this collection of modifications: Those who are interested in only optimizing Wayne's design, and those, for whatever reason, want more output than the original design is capable of providing. Because of this, I will present the modifications from easy optimization, through "full tweak" optimization, to "beyond original design" modifications. So each potential modifier has an idea of return versus time invested, here is a table of my results (13.8V supply except for the last column).

BAND	INITIAL OUTPUT	FIRST CHGS	REDUCED PI INDUCT.	NEW PA	ALL CHANGES W/15.2V P/S
80	built initially		4.2W	4.2W	4.3W
40	2.5W	2.6	4.0	4.9	5.8
30	2.1	2.2	3.0	4.0	4.8
20	1.8	1.9	2.6	3.2	4.0
18	1.0	1.3	2.0	2.9	3.5

As you can see, the initial gains are small, but I recommend doing them before doing the more major changes. First, lets cover the modifications that Wayne has called for since release of board revision C.

1. Replace Q5, a J309, with a J310 which is "hotter"

2. Make sure you *really* have the trimmers peaked. This means making sure you re-peak C66 & C64 for max power out after peaking the XMIT BPF. Do your

trimmers all have 2 peaks in them as they are rotated 360 degrees? If not, remove or add inductance to the associated toroid until they do.

3. Place a ferrite bead, like a FB101-43 or FB101-64, on the base lead of the 2N2222A driver. This will both increase power 15M and up, and allow you to build 12M & 10M modules now offered by Wilderness Radio.

4. Ensure that the T2 primary winding has 11 turns, not 8 turns.

5. Replace the Output Filters, C47, C48 & C49 with Silver Mica, or Polystyrene capacitors. Large value Silver Micas are hard to fit on the board, so use Polystyrene for the lower band modules. I used Meacham's "Optimized Sierra Output-Filter Values" published in QRPp Sept. 95, but using his values requires re-winding most of the L5 & L6 Toroids. If you don't want to be bothered with tearing apart all your modules, certainly do the higher frequencies, from 30M and up.

6. Finally, change the zener protection diode, D7, to 43V instead of 36V. Also change R15 to 36 ohms to avoid exceeding the base-emitter breakdown voltage of Q7 due to excess drive. These changes are important if you run higher supply voltages.

With modification 6, you now have the ability to control output power by varying the supply voltage. Wayne suggests 12V in the field for reasonable battery drain, and 16V on the desktop for an approximate 50% increase in power output. Now, some unpublished final tweaks from Wayne for those that don't mind some work for 1/10 of a watt gain! (sample of one)

Wayne's instructions will take you through the Sierra TX strip, from the beginning to the Final to ensure proper signal levels:

1. Premixer

The pre-mixer has two inputs, VFO and crystal oscillator. 1A. Try putting another resistor in parallel across R22. Start with about 4.7K. This will increase VFO injection into the premixer. See if final output power goes up, stays the same, or goes down. If it goes down, you may actually have too much injection into the premixer, and should try reducing the size of R22.

1B. The crystal oscillator injection may be low. You can do two easy things to test this:

i. Vary C70 on the band module and see if power goes up or down. You may have had it set or very low capacitance, reducing oscillator level.

ii. Put another 47K resistor across R9. This increases oscillator bias.

2. TX mixer injection

There are two sources for mixer injection. The premix buffer output and the TX 4.915MHz oscillator.

2A. Parallel another 5pF cap across C29. Note if this makes power output go up, stay the same, or go down. If it goes down, injection may be too high, and you may need to reduce the size of C29.

2B. Try adding a 22K resistor from U8 pin 7 to ground. This will increase TX mixer oscillator output.

By making all these tests, I found my Sierra was close to optimum, but with a little padding here and there, I picked up 1/10 Watt. Unless your Sierra is not performing "up to spec", I would not suggest these tests unless you really enjoy tweaking.

The next modification requires some explanation and some qualifications. As Dave Meacham pointed out in his "Variable-Power Caveat" article in QRPp, Sept. 95, The output filters for the PA are designed for a specific termination resistance, usually 50 ohms. When one changes the collector voltage, the filter becomes mismatched to the PA, which, among other things, may reduce the attenuation of harmonics. Keep in mind that most qrp rig designers give you a drive control for the express purpose of adjusting the power

output. When you crank the power down to 1 watt, you are also making the same kind of mismatch to the PA output filters. For any given power level therefore, there is an optimum value for the output filter.

Having stated Dave's warning and ignoring it, I modified the LPF filter by reducing inductance on each module's L5. Why? Because early Sierra tweakers found that the PA really would like to see about 35 ohms, not 50 ohms. Wayne gives Bob Warmke credit for being first to report power increase by tweaking his LPF inductors. I removed 2T from L5 on 80 & 40M, and 1 T on the rest. You can see from my chart that this change represents the largest power gain I found.

Wayne, however, states that the right way to adjust the matching is to replace the collector choke with a 2:1 transformer. He says that he knows at least two Sierra builders that have done this, but further states that "the only problem may be that the higher bands roll off even more now than they did before". I have not tested his recommended approach.

As the last caveat for this particular modification, no one, to my knowledge, has run harmonic output tests to let us know what changing the PI filter to the PA collector does. Wayne's guess is "It's liable to be significantly different, most likely worse, but probably legal".

The final modification, clearly not part of the original design, is to replace the PA. In my case, I was on a roll with all these tests, and egged on by Preston Douglas and Bob Gobrick. I ordered a MRF237 from MCM for $11 plus change. No components changes are needed for this swap, but the pin-outs are reversed. This means you need to fold the base lead across the case, without touching it, and place its offset with the E & C on the other side of the case. (You have reversed the E & C connections). No other changes are required, though you might consider a "high output" heat sink. Some correspondents thought I might see a small change, Wayne predicted "0 to 20% higher output depending on the band", but as you can see by my chart, I picked up substantial power on several bands.

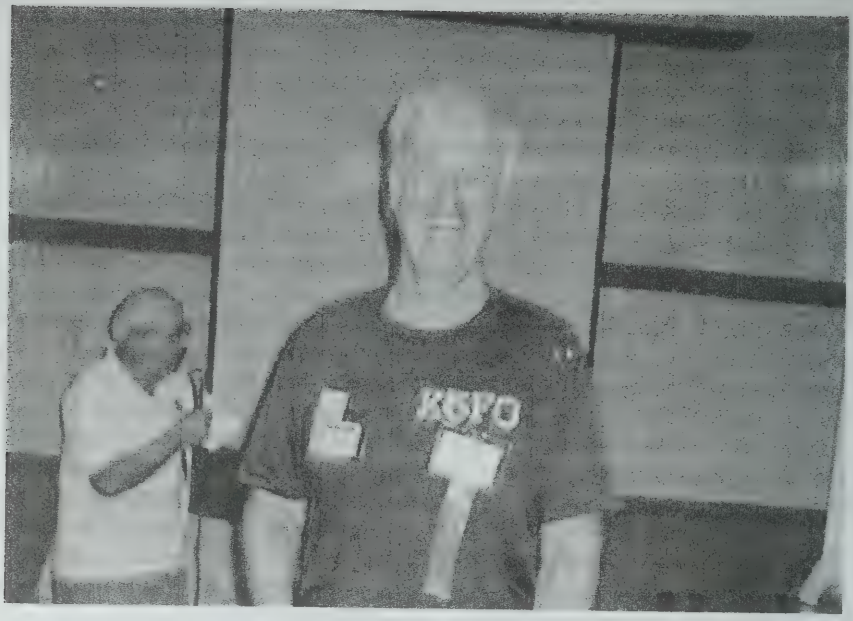
There you have it. A full range of tests and modifications from which you can choose to increase Sierra output. But again, do keep in mind that the easiest way to increase power is to increase supply voltage. That, in conjunction with several of these modifications give you a wide range of qrp levels to chose.

Pacificon 96: The First Annual West Coast QRP Symposium

by Doug Hendricks, KI6DS
862 Frank Ave.
Dos Palos, CA 93620
ki6ds@telis.org

Jim Cates and I sat down after Pacificon 95 to discuss NorCal's participation in the event. We had a booth, and we sponsored 2 QRP forums, one on Saturday and one on Sunday. Jim suggested that we expand our presence in 1996, and one of the ideas that he came up with was to fly in several more speakers from other areas and to have a "West Coast QRP Symposium". We had such a suggestion from one of our speakers, Derry Spittle, VE7QK. Derry had mentioned that Dayton was a long way for QRPers on the west coast to travel to, but most of them could get cheap air fares up and down the coast.

We decided to contact the Pacificon people and see if they would be ameable to the idea. We would require 6 sessions, plus a room to host the QRP Open House session on Saturday night. I contacted



Chuck Adams, K5FO at Pacificon

Dick Brown who put me in touch with Greg Estep, who was in charge of the sessions for the 1996 Pacificon ARRL Pacific Divison Convention.

Greg listened to our proposal, but was not too enthusiastic, as the number of rooms and sessions were full, and it didn't seem to be possible for us to have the 4 extra sessions and the open house. We were disappointed, but understood. Then, about a month later, Greg contacted us and said that we could have 4 afternoon sessions if we were willing to use the room that was vacated by the VE testing team at noon. The room would seat about 100, and it was perfect for our needs.

The West Coast QRP Symposium was on! Now we needed to contact the speakers and get them confirmed. We wanted to provide a variety, as we have learned that one thing that our members seem to like is variety. Jim and I talked and decided on the following speakers; Chuck Adams, K5FO, Eric Swartz, WA6HHQ, Stan Goldstein, N6ULU, Derry Spittle, VE7QK, Ed Burke, KI7KW, Bob Tellefsen, N6WG, Paul Harden, NA5N, and Doug Hendricks, KI6DS.

The speakers all agreed to come and we were in business. All that was left to do was to arrange transportation and hotel accomadations, which was simple to do.

We advertised the event on the internet via the QRP-L group and the NorCal Web Page and in the September issue of QRPp. The NorCal local members made plans to attend, and we had several out of area members who made arrangements to attend. I received several messages on the internet, and felt that the event would be a success. But I had no idea as to the response we would have. Every session was standing room only, and very well received. All of the attendees commented that the QRPers were sure having a lot of fun in their sessions. This was due to the humor in the presentations and QRPers natural tendency to have fun.

The first presenter was Chuck Adams, K5FO. Chuck is perhaps the most recognizable QRPer in the United States today due to his involvement in so many

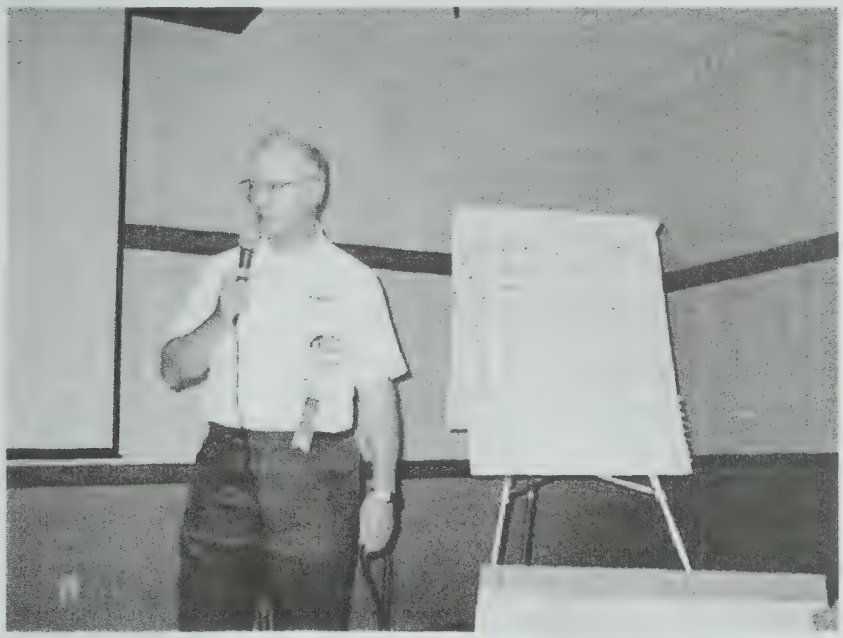
QRP Activities. Chuck writes many articles for QRP journals, was the founder of the QRP-L group on the internet which now has over 1300 subscribers, and also started the Fox Hunt contest and the 30 and 80 meter propagation study.

Chuck Adams may be the perfect person to open a QRP Symposium. He is a very entertaining and personable speaker who knows his subject extremely well. Chuck gave the audience a brief history of QRP and then gave us a picture of where we are today and where he thinks that we are headed in the future. Chuck's main message was that it was iimportant for all QRPers to get on the air and operate to increase their skills, improve their knowledge and advance the state of the art.

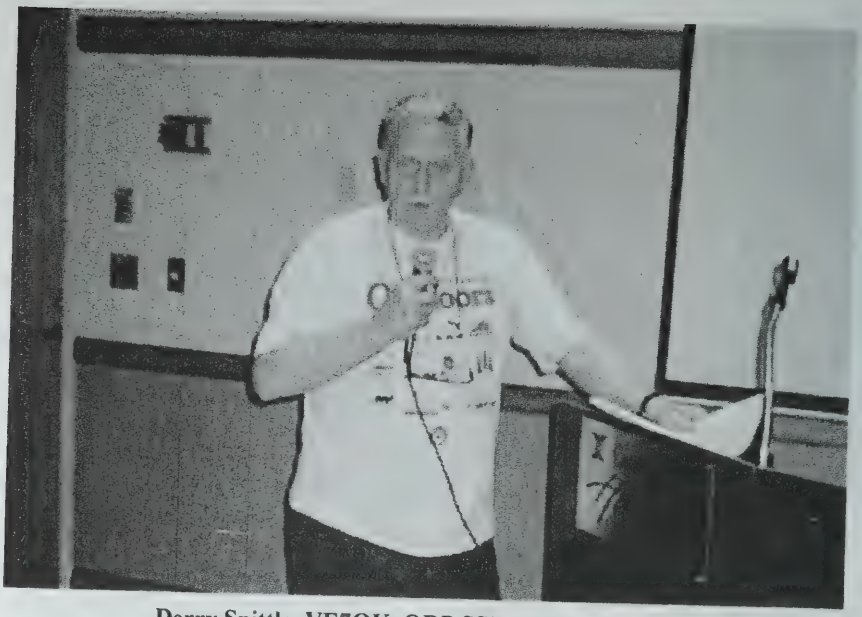
The next speaker was Bob Tellefsen, N6WG, who may have been the best kept secret in QRP Field Day circles before the WCQRP Symposium, but now the cat is out of the bag. Bob is the leader of the Alemeda County FD group that has won their category for the past 11 years! Bob gave us the secrets and hints as to how his group made 1253 QSO's at the bottom of the sun spot cycle this past June.

Bob's secrets include using female operators on SSB, having a full contingent of operators for every rig, and keeping them staffed at all times, planing, having pre-cut and built antennas so that all you have to do is put them up, and making sure that everyone is familiar with the rigs that they are going to be using.

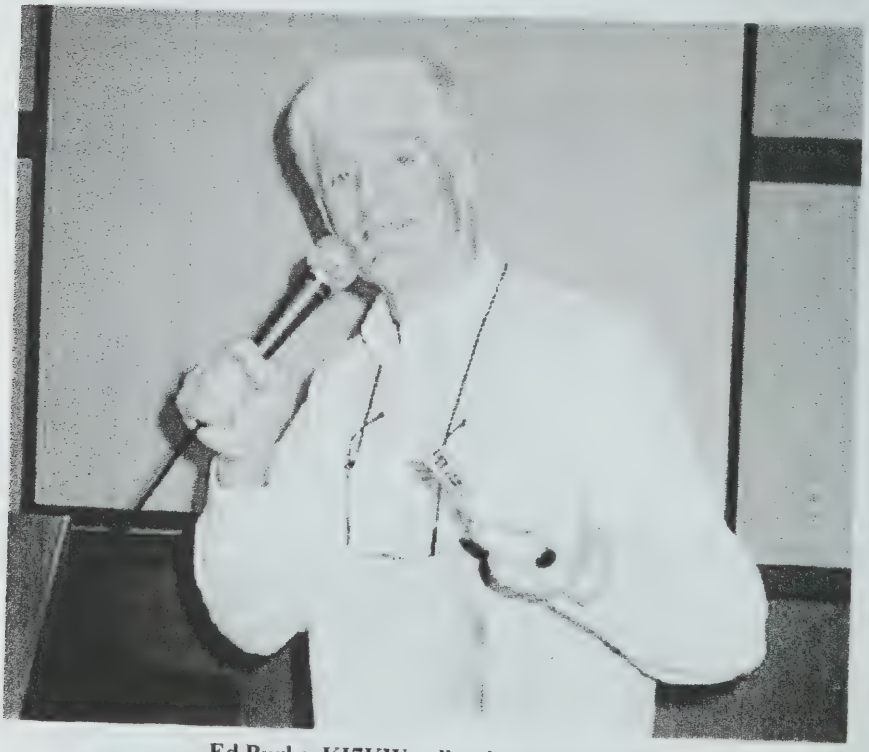
Derry Spittle, VE7QK, and Ed Burke, KI7KW are two of the better known builders and designers of SSB QRP rigs on the west coast. Derry is from Vancouver , British Columbia and Ed lives near Portland, OR. They routinely carry on a nightly schedule on 75M with the homebrew QRP rigs that they have built. They shared their experiences in solving some of the problems of QRP SSB operation. The surprising message that came out of their presentation was that 75 Meters was a better



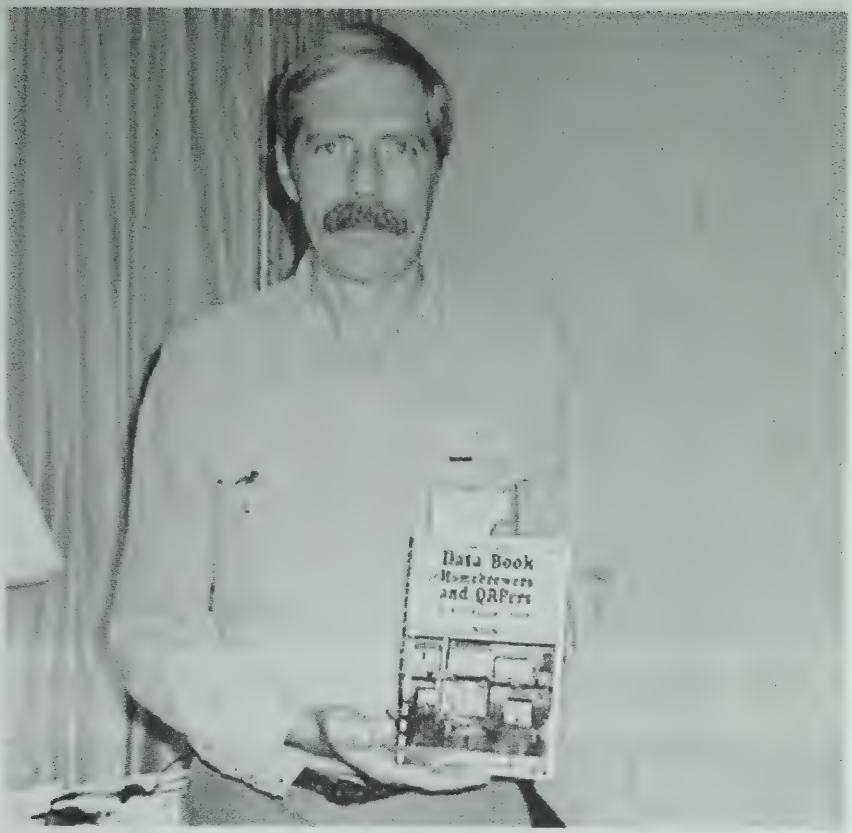
Bob Tellefsen, N6WG "Preaching to the Choir about QRP FD, Pacificon 96"



Derry Spittle, VE7QK, QRP SSB Pioneer Designer



Ed Burke, KI7KW, talks about QRP SSB



Paul Harden, NA5N and his new book, Handbook for QRPers & Homebrewers

band for QRP SSB than 20 Meters

Paul Harden, NA5N, from Socorro, NM was flown in for the conference to repeat his presentation at Dayton for the NorCal Open House during FDIM. Paul's presentation was a true work of art. He explained how the designers design the various features into the QRP rigs that we are all so familiar with. Paul had a handout that featured the schematic of the NorCal 40, MFJ 90XX series, and the NW 8040 rigs. He then explained to the audience how each designer used different and similar circuts to make the radios work.

The final forum presentation on Saturday was a panel of DX and Contest QRP operatiors featuring Stan Goldstein, N6ULU, who has DXCC #1 with a NorCal 40 and now has 130 countries with the NC40. Stan also was the first year Fox champion in the fox hunt, and then last year he won as a hound in the second year of the contest. Eric Swartz, WA6HHQ, was the second member of the panel, and he has plenty of credentials too. Eric has NC40 DXCC #2, and works just about every DXpedition that comes up by using QRP and a great antenna system. Chuck Adams, K5FO was the third member of the crew, and rounded out the panel.

The audience was again full as the NorCal members listened with rapt attention to Chuck, Stan and Eric talk about how they find DX, work it an the tricks that they use to do so. Many secrets were shared, some of them surprisingly simple, yet effective.

We took a two hour break after the

Chuck Adams, K5FO, Eric Swartz, WA6HHQ, & Stan Goldstein, N6ULU
1996 West Coast QRP Symposium Contest and DX Panel

Dennis Utley; Jim Cates, WA6GER, Monte Stark, KU7Y and the crowd.

The 49er Contest winners. L-R: Doug Hendricks, Darrel Jones, Mike Gipe, Vern Wright, Wayne Burdick. Front L-R: Jessica Gipe & Winner Debra Blanke.

afternoon session for dinner, and then all of the Symposium attendees returned for a QRP Open House/Show and Tell session at the hotel. This was the first time that we have done the symposium, and Jim and I wanted to try and capture the flavor of the ARCI sponsored hospitality rooms at Dayton.

It didn't take long for the members to get into things. When I walked into the room at 7:00, it was full of QRPers, all talking in small groups, sharing ideas, and proudly showing off their latest projects. The speakers were all there, and it was a wonderful opportunity for the members to get to meet and know the "famous QRPers".

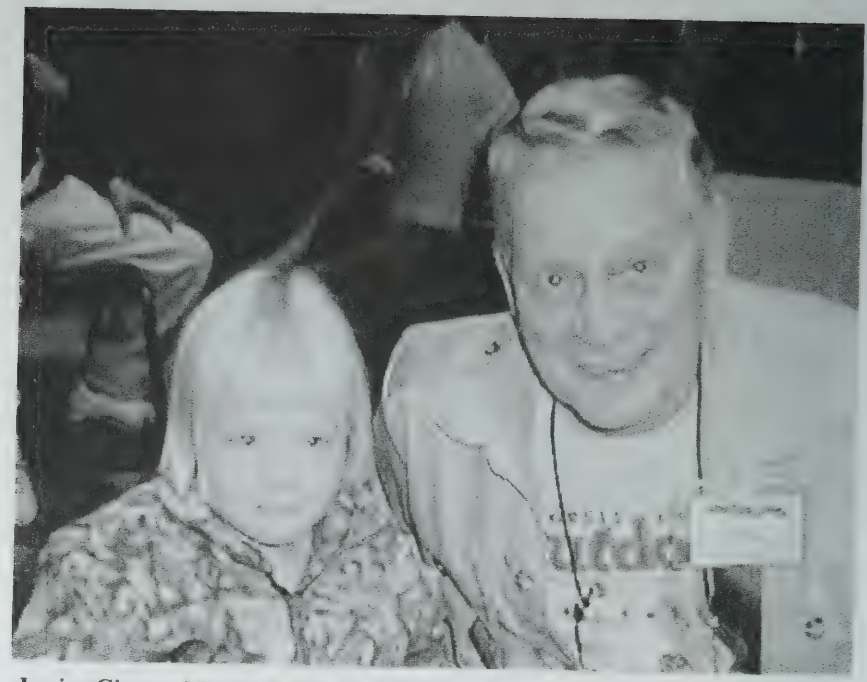
One of the feature attractions of the open house was the West Coast edition of the 49er building contest. Their were 6 entries this year, and all were very well done. The results were:

1st	Debra Blanke
2nd	Vern Wright
3rd	Mike Gipe
4th	Darrell Jones
5th	Jessica Gipe (paper 49er)
6th	Doug Hendricks

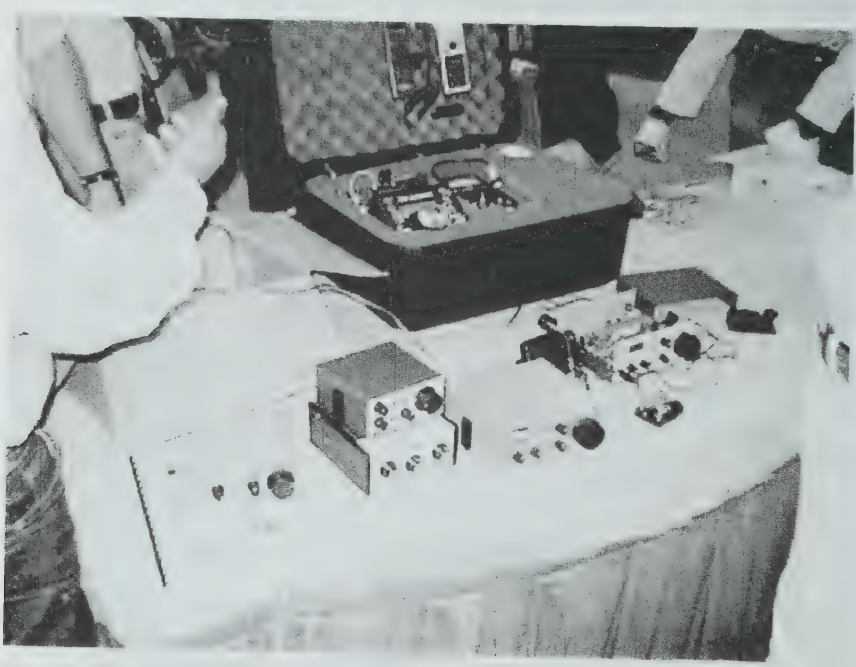
The first 4 places won new SST transceiver kits that will be the next NorCal club project and is being designed by Wayne Burdick, N6KR. Prizes will be awarded when the kits are available. The SST will be featured in the next issue of QRPp. Jessica deserves special mention,as she is only 5 years old! She was watching her Dad build his entry and decided to enter the contest too. Her prize were two bags of M&M's, plain and peanut, as the judge was not sure of her preference.

Chuck Adams, KF50 did a great job of judging, and has promised to return next year to judge the "38 Special" and Rainbow Tuner entries. I don't know what time the party broke up, as I went to bed at midnight and it was still going strong.

Sunday morning, I was the speaker for the 10:00 AM session, which has usu-



Jessica Gipe and Derry Spittle, VE7QK shared QRP stories at the Open House



A small portion of the many rigs on display

Monte Stark, KU7Y, Editor of the QRP ARCI was a special guest at the NorCal West Coast QRP Symposium.

ally been sparsely attended. My subject was how to get started in QRP. I was pleasantly surprised to see a very good crowd in attendance. QRP is really growing. I explained what QRP is, and how easy it is to do.

One of the main things that I tried to impart was that it is fun and easy to build. Many new builders are afraid to build because they have heard two myths of building; one that parts are hard or next to impossible to find, and two that toroids are horribly difficult to wind. I explained that the kits suppliers today have eliminated the first problem, and gave a toroid winding demonstration using a roll of toilet paper as a toroid and some RG174 as the wire to get rid of the second myth.

After the presentation, we all said goodbye to our friends and made plans to attend next year. Jim and I both agree that the West Coast QRP Symposium was a lot of fun and well worth doing again next year. Our emphasis is on having fun, and doing things the right way. We feel that we succeeded and hope that you will be there next year to share in the fun. Mark your calender for the 3rd weekend in October as the time to meet in Concord, California for QRP fun. We will bring another group of quality speakers from throughout the country. On behalf of Jim and the NorCal membership, I would like to thank the speakers who gave of their time to give us outstanding presentations.

Many people have asked me why QRPers seem to have so much fun when they get together. I don't know for sure, but I think the reason is that QRPers do things, and they don't mind helping others. We certainly proved that at Pacificon.

Let me close by saying thank you to Jerry Parker and Dennis Utley for the great photos in this article. See you next year. 72, Doug KI6DS

Figuring Sheet Metal Allowances

by Vic Black, AB6SO

If you've ever tried to bend up a sheet metal box or cover for one of your projects you may have wondered why the part didn't fit after bending. Maybe you forgot to include an allowance for the bend. A galnce at Figure 1 shows that when a piece of sheet

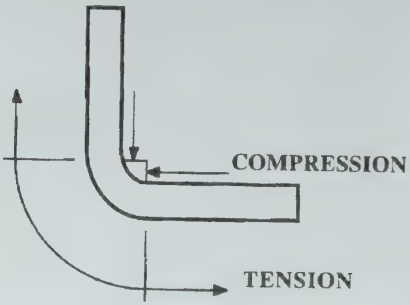


FIGURE 1

metal is bent, the inside surface is shortened (or squeezed) by compression and the outside surface is lengthened (or stretched) by tension. Someplace in between is a neutral axis where neither shortening nor lengthening takes place. The sheet metal in the bend is true length only along the neutral axis.

The total length of sheet metal must be calculated. An example of a simple angle bracket is shown in Figure 2. The bracket is to be 1 inch long on each side with a bend radius of 1/16 inch (0.063) and metal thickness of 0.040 inch.

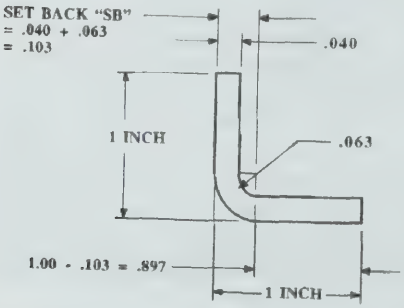


FIGURE 2

The first step is to figure out the set back (SB) which is the sum of the bend radius (BR) and the metal thickness (MT). In this case the set back equals .063 +.040 = .103 inch. The set back is subtracted from the overall side dimensions to find the lengths of straight, unbent legs of the bracket.

The next step is to add the bend allowance (BA), or length of the neutral axis in the bent portion, to obtain the overall length of the flat pattern for the bracket. This is the tricky part. We'll digress for a bit and discuss the geometry of curves.

The circuference of a circle is equal to 2piR. Since there are 360 degrees in a circle, it follows that the circumferential length for one degree is equal to (2 pi R)/ 360 or (pi R)/180. Solving for (pi R)/180 we get (0.0174 x R) inch per degree of bend on the **inside of the bend** (that's on the compression side). Since sheet metal has a finite thickness, a constant must be added to move out to the neutral axis. We'll use 0.0078 times the metal thickness. this gives us a final formula for the bend allowance as follows:

Bend Allowance = BA = (angle of bend) x [(.0174 x BR) + (.0078 x MT)].

Solving the formula for our example above (Figure 2), we get:

BA = (90) x [(.0174 x .063) + (.0078 x .040)]

BA = (90) x [(.0010962) + (.000312)]

BA = (90) x (.0014082)

BA = .127 inch

This makes the final flat blank look like Figure 3.

The calculations may look complicated, but a simple 4-function calculator

with a memory makes them easy to do. Incidentally, since this method is based on the geometric properties of the shape, this method works regardless of what dimensioning system you use so long as all dimensions are in the same system. That means that you can use metric rather than English dimensions in the formula and still get the same results.

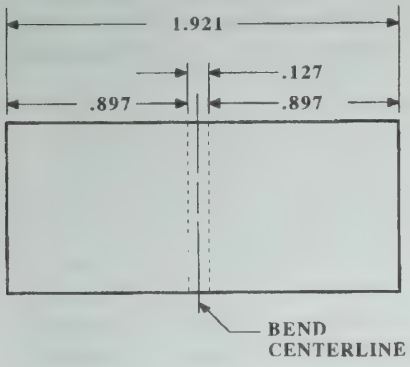


FIGURE 3
(FLAT DEVELOPMENT OF FIGURE 2)

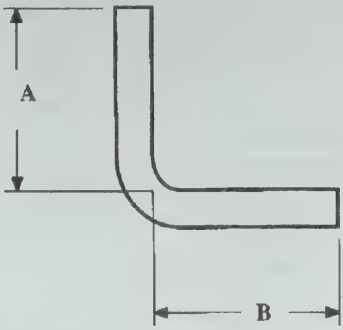


FIGURE 4
(ALTERNATE METHOD)
TOTAL FLAT LENGTH =
A + B + CORRECTION FACTOR

KC-2/Cascade Installation

by Dave Meacham, W6EMD
206 Frances Lane
Redwood City, CA 94062
ddm@datatamers.com

Coupling the KC-2 to the VFO in the usual manner doesn't work well in the Cascade. The Hartley circuit seems to be more susceptible than the usual Colpitts to a pulse signal that comes OUT of the KC-2 at the VFO input. That signal is the KC-2's "10Hz VFO-sample-interval timer" pulse. It modulates the VFO, causing a "warble" at 10Hz. The following mod solves the problem.

Changes for the Cascade/KC2 installation:
1) Change C8 in the KC-2 to 100pF.
2) Run RG-174 from the KC-2 to U7 (I use about 4.5 inches of it).
3) Couple the center conductor to pin 7 of U7 with a 15pF C0G or NP0 cap.
4) Ground the coax near U7. Leave the coax shield at the KC-2 end disconnected.

(This mod adds an additional buffer between the KC-2 and the VFO.)

In normal SSB operation I don't hear any evidence of the 10Hz pulses. If I zero beat a strong carrier I hear it weakly (also near zero beat). Wayne pointed out that the AGC is not working under these conditions, so there is lots of gain. I'm happy with it as is, and its a simple mod.

KC-2 Connector for J3

I am using a connector that fits the KC-2 (J3) very well. Thought I'd pass along the Digi-Key part numbers:

Item	Digi-Key Part Number
18-pin header	S2011-09-ND
18-pin-connector housing	WM2526-ND
Female "pins"	WM2513-ND

The only problem, a minor one at that, is that there is no locking gizmo nor any other kind of polarizing thing. A tiny dot of red nail polish on the housing and on the header should do the trick.

72, Dave, W6EMD

XIT Mod for the Sierra

by Eric Swartz, WA6HHQ
eric@cruzio.com

I mentioned during my talk at Pacificon that I used XIT on my Sierra to help work DX stations. Many have asked for a copy of the circuit, so here it is. Enjoy.

1. Remove S1, and use S1 holes for points "B" and C".

2. Add S2, (Note: up position uses lower pins.)

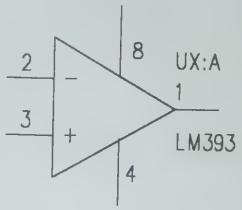
S2 up = XIT
Center = Off
S2 Down = RIT

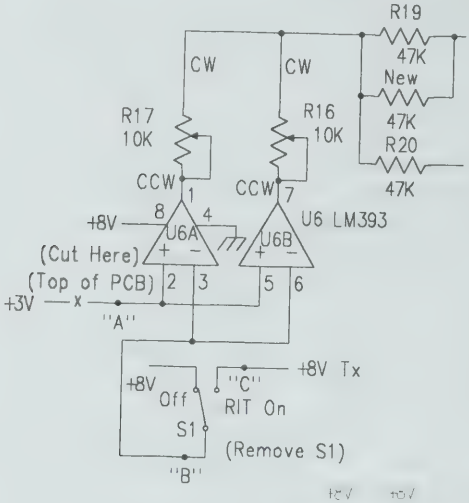
3. Add a 47K resistor in parallel with R19 to increase the RIT/XIT tuning range.

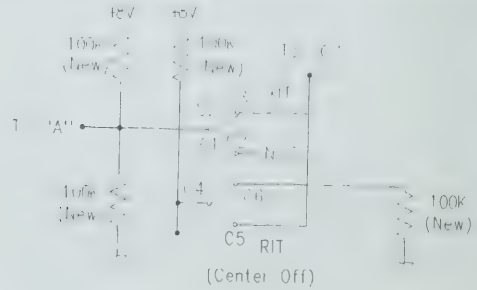
4. Change R16 to a 10K Pot (set at mid range.)

(Note: changing R16 to a 10K Pot allows moving the center of the RIT/XIT range. I typically set it to + 6 kHz, -1 kHz at band bottom).

Notice that the LM393 comparator has an open collector output. When Pin 2 (-) voltage is greater than pin 3 (+); output is Grounded. When Pin 3 (+) voltage is greater than Pin 2 (-); output is Open.







Convert a 49er to a 39er

by Jeff Anderson, WA6AHL
1607 Bonita Ave.
Mountain View, CA 94040
jeffa@ix.netcom.com (Jeff Anderson)

Here is a simple way to convert a stock 49er kit to operation on 30 Meters. Have fun.

C17: change from 270 pf to 470 pf
C18: change from 470 pf to 560 pf
C20: change from 150 pf to 68 pf
C21: change from 5 pf to 22 pf
RFC5: change from 2.2 uH to 0.82 uH
RFC6: change from 15 uH to 10 uH
X1: change from 7.040 MHz to 10.116 MHz

That's it. RFC5 & RFC6 are molded "solenoid" inductors similar in style to those currently used in the 40-9'er. 73, - Jeff

FYBO Winter QRP Field Day

by Joe Gervais, KC7NEV
PO Box 1822
Goodyear, AZ 85338.
vole@primenet.com

Hear Ye, Hear Ye!

Here it is folks, the formal announcement of the event which will prove once and for all that "QRP is not for Sissies." Yes, it's the FYBO (Freeze Your B____ Off) Winter QRP Field Day, sponsored by the Arizona ScQRPions.

1) The words "winter" and "field day" may not seem compatible to some of you. You're thinking of staying home by a warm fire and listening to your Louis Armstrong collection while picking off QSO's. That's fine! But please consider just a few hours in "the field". A park, a patio on the side of ski lodge, whatever. Bring a warm coat, some hot chocolate, the spouse and kids, the dog, whatever helps make it fun and comfortable. Build a snowman to make it a multi-op station. Whatever! (We southern Arizona types have to build our snowmen from tumbleweeds - oh the hardship!)

2) To reward those who venture into the great outdoors (or those with unheated ham shacks), we've added a temperature multiplier. Please note that this is for the temperature AT THE OPERATOR'S POSITION. Stick a thermometer by your keyer, and there you have it. Indoor operators report indoor temperature. Please be fair.

3) We've tried to benefit the Novice/Tech Plus types. One common problem is that nobody spends time in the Novice portion of the band, because there aren't enough signals there, because nobody spends time there, because.... You get the idea. So, all QSOs made in the Novice sub-bands are worth extra points, regardless of who you contact. But if you actually work a Novice/Tech, you get big points. So get ahold of any Novice/Techs you know, sit 'em in front of a QRP rig, and hopefully they'll end up having the time of their lives. This is an experiment that I hope will work.

4) The key word here is FUN. To that end, and to spice things up for all, we've added a random drawing from all logs received. This way even the "little guys" with marginal antennas or rusty skills have a chance at winning something. And that something is a Ten-Tec QRP Xcvr kit, band of your choice. But logs will be cross-checked where possible, so no fair faking one if you don't participate.

Remember, "If it ain't fun, you aren't doing it right!" Cheers de KC7NEV, Joe, vole@primenet.com, AZ ScQRPions #7

QRP CONTEST ANNOUNCEMENT

FYBO Winter QRP Field Day

Sponsored by the Arizona ScQRPions

**** SAFETY FIRST! Please respect the weather and your own limitations. ****

When: 1700Z Feb 22 - 0500Z Feb 23
QRP Only.
Modes: CW, Phone.
Bands: HF, standard QRP calling freqs (no WARC). For Novice portion of bands, suggest 3.710, 7.110, 21.110, 28.110. Work stations once per band.
Exchange: RS(T), state/province/DXCC country, power output, and temperature (Fahrenheit) at OPERATOR`S POSITION. Indoor stations must report indoor temperature.
Example: "RST 579 AZ 2W 58F". Novices sign with /N, Techs sign with /T.
Points:
- 5 pts/QSO with a Novice/Tech ham (i.e. contact a Novice/Tech, get 5 pts).
- 2 pts/QSO in Novice portion of bands with General class or higher ham.
- 1 pt/QSO all other.

Example: W1AW (Extra) completes QSO with KD6PRD (Novice).
W1AW gets 5 pts. KD6PRD gets 2 pts.

Multipliers:
- States/provinces/DXCC countries
- Field Location = x 4
- Alternative Power Source (battery/solar/wind/etc) = x 2
- QRPp (less than 1w) = x 2
- Lowest temperature recorded at OPERATOR`S POSITION while on the air:
  60+ F = x 1
  50-59 F = x 2
  40-49 F = x 3
  30-39 F = x 4
  20-29 F = x 5
  Below 20 F = x 6

Final Score:
QSO points x Multipliers.

Awards/Prizes:
- Lowest operating temperature.
- Most Novice/Tech stations worked.
- Highest score (single op, multi op, Novice/Tech).
- Random drawing from logs received.
- Send 9x12 SASE with log for certificate.

Email vole@primenet.com for more information.
Send logs by Mar 21 to Joe Gervais, KC7NEV, PO Box 1822, Goodyear, AZ 85338.

QRP HINTS & KINKS

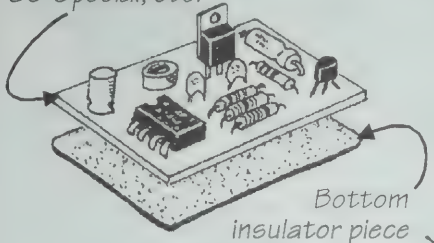
A NorCal Exclusive

Illustrated by Paul Harden, NA5N

#2 USING HOUSEHOLD ITEMS

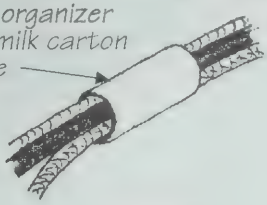
USES FOR MILK CARTONS

Breadboard, 49'er, 38 Special, etc.

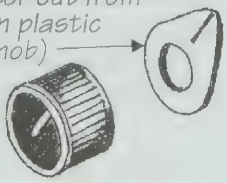


Bottom insulator piece cut from a milk carton

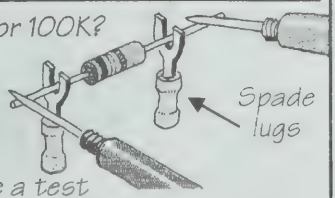
Cable organizer from milk carton handle



Knob pointer cut from milk carton plastic (Glue to knob)



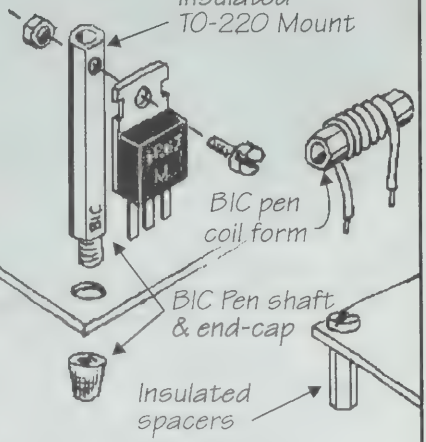
10K or 100K?



Make a test saddle by mounting spade lugs on solid surface; makes testing components quick & easy.

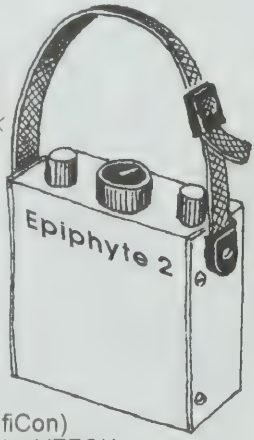
From Vic Black, AB6SO

USES FOR "BIC" PENS



EQUIPMENT HANDLES ...

can be made from cloth or nylon adjustable straps sold at many clothing stores, sporting goods, and at stores specializing in camping and hiking supplies. Also back-pack replacement straps which includes mounting brackets and slip buckles.



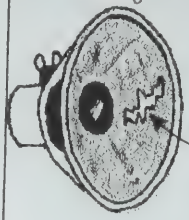
(As seen at PacifiCon)
From Derry Spittle, VE7QK

More QRP Hints & Kinks

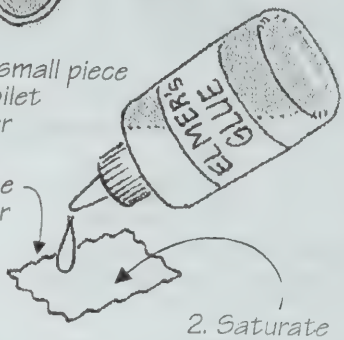
Some SPEAKER Hints & Kinks

HOW TO REPAIR A TORN SPEAKER
(In the privacy of your home)

This technique works quite well in repairing torn speakers. Some low frequency response might be lost, but hardly noticeable for CW or SSB reception.



Rip, tear, puncture wound, etc.



1. Cut small piece of toilet paper

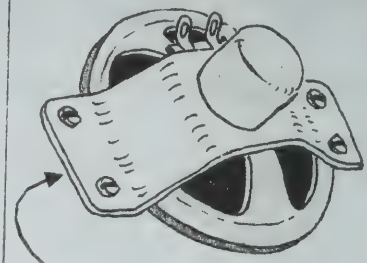
tissue paper

2. Saturate with wood or paper glue

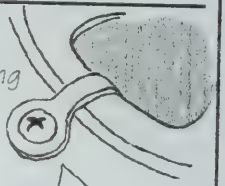
3. Place soaked tissue paper over damaged area like a bandage

4. Repeat for other side of speaker if the damage is severe

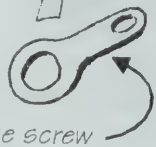
MOUNTING ROUND SPEAKERS (no flanges)



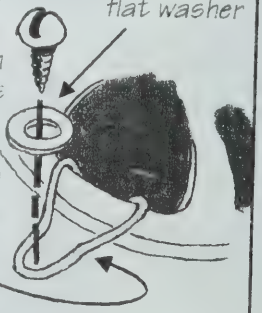
Form mounting "saddle" from milk carton plastic or thin metal from a "soup" can.



Form mounting clips from solder lugs. Be careful to not "poke" the speaker cone.



#6 Solder lug for #6 woodscrew or 6-32 machine screw



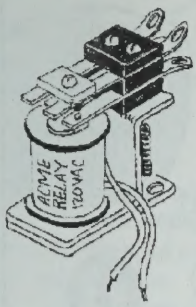
Form 3-4 mounting clips from solid buss wire and secure with wood screws or machine screws.

#6 or #8 flat washer

Send your ideas to:
Paul Harden, NA5N (NA5N@Rt66.com) or Doug Hendricks, KI6DS

More QRP Hints & Kinks

WEEKENDER PROJECT --
Build a nifty cheapie set of keyer paddles



This set of paddles is made from the contacts of an AC relay or old phone jack. Contacts are rearranged, mounted on a base, & secured to table top with suction cups.

Inspite its simplicity, they work well and have an excellent feel!!!

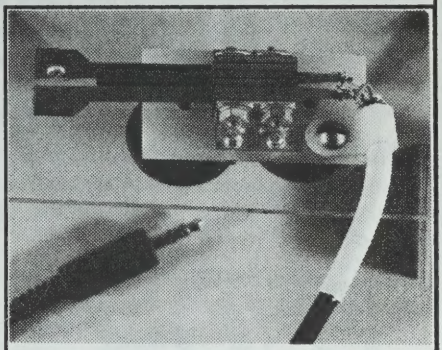
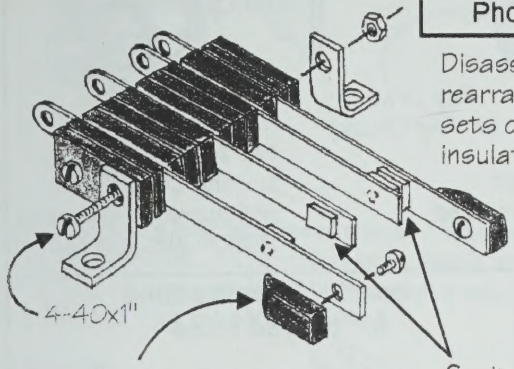


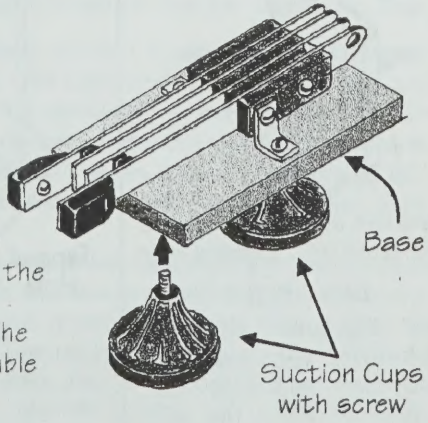
Photo of W6MMA's paddles



Disassemble relay contacts and rearrange as shown, using two sets of contacts and the insulating wafers. Bolt together with long 4-40 machine screw and nuts. Place bracket on one set of bolts for mounting to base.

Form "paddles" with piece of plastic, wood, rubber, etc. to shape & size of preferrence.

Mount completed unit to an insulated base (wood, fiber glass, plastic, etc.) with the brackets. Mount suction cups by drilling and tapping a hole in the base to match threads on the suction cups. Heigth can be adjusted to suit one's "feel." The suction cups secures key to table top for long periods of time.

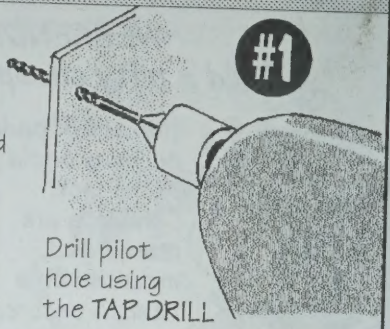


Submitted by: Vern Wright, W6MMA, Placerville, CA

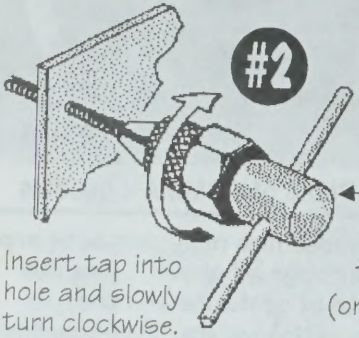
QRP Hints & Kinks

WORKSHOP PRACTICES
Tapping Holes

Buying a couple of tap drills, taps and a tap driver can greatly enhance your metal-working capabilities. The most common sizes for electronic work are 4-40 and 6-32. Drills are $1, taps $2 and a tap driver $5-10.



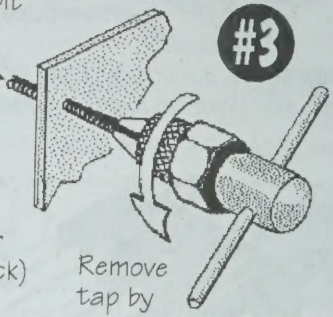
Drill pilot hole using the TAP DRILL



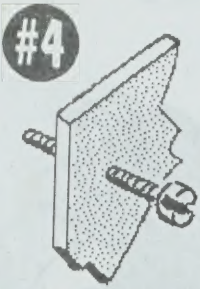
Insert tap into hole and slowly turn clockwise. Tap will thread the hole. Use proper tap drill ... a smaller sized drill can cause tap to jam and snap-off.

Tap Driver (or Tap Chuck)

Tap Bit



Remove tap by turning CCW. Remove any burrs on far side that might jam tap.

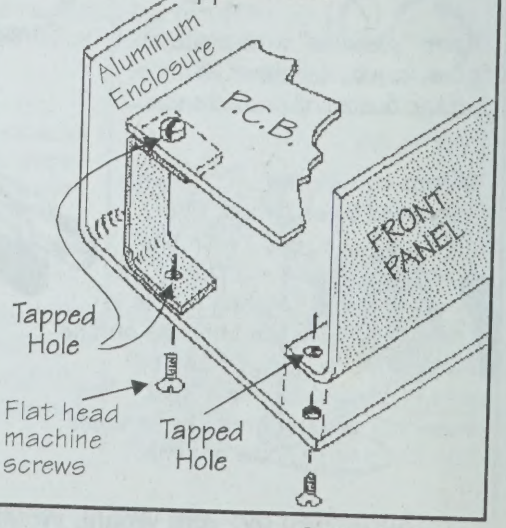


Machine screw can now be threaded into tapped hole.

DRILL SIZES

Size	Tap Drill	Thru Drill
2-56	.070" #50	.086" #44
4-40	.089" #43	.116" #32
6-32	.107" #36	.144 #27

A couple of applications for tapped holes



Back Issues of QRPp

Back issues of QRPp are available in bound issues only. Please do not request individula issues, as they are not available. The back issues are printed as sets, and it is just not possible to break up sets.

				Price	Shipping
1993	Volume I	3 Issues	140 Pages	$10.00	$3
1994	Volume II	4 Issues	296 Pages	$15.00	$3
1995	Volume III	4 Issues	278 Pages	$15.00	$3
1996	Volume IV	4 Issues	284 Pages	$15.00	$3

DX shipping is $5 per issue, with 2 or more issues $10, shipped to Europe, Africa and South America airmail. Shipping to Asia and Australia is $5 per issue, with 2 or more issues $10 shipped surface. Airmail shipping is $10 per issue, with 2 or more $20 for Asia and Australia.

All 4 issues, 1993-96, $60 postage paid, shipped priority mail in the US. $65 Europe, $75 Asia and Australia, Airmail. US funds only, please make checks and moneyorders out to Doug Hendricks, NOT NorCal. Send orders to:

Doug Hendricks, KI6DS
862 Frank Ave.
Dos Palos, CA 93620

Crystals

NorCal has made a bulk purchase of crystals. We now have the following frequencies available: 7.040 MHz., 7.122 MHz (Novice Band), and 10.116 MHz.(30 Meters) the price is $3 each, postage paid. Make checks or money order out to Doug Hendricks, 862 Frank Ave., Dos Palos, CA 93620.

QRPp is published at Dos Palos, California 4 tiems per year; March, June, September and December. Subscriptions fee is $10 for US addresses, $15 for Canada and $20 DX per year. To join the NorCal QRP club, send your name, call sign, and address to: Jim Cates. There is no charge for membership to NorCal QRP Club. To receive QRPp, you must subscribe and pay the fees which entitle you to receive 4 issues of QRPp. Send your money (US FUNDS ONLY) to: Jim Cates, 3241 Eastwood Rd., Sacramento, CA 95821. Please make checks and money orders out to Jim Cates, NOT NorCal.

The articles in this journal have not been tested nor is there any warranty as to the feasibility of the items described. The articles have been published with the consent of the authors, and no article may be reprinted or reproduced in any form without the expressed written consent of the author. All authors retain all copyrights to their materials, and all articles in this publication are copyrighted. Publishers of other club newsletters may reprint articles as long as they are NOT for profit and not a commercial venture of any kind, and credit is given to the author and QRPp.

QRPp, Journal of the NorCal QRP Club
862 Frank Ave.
Dos Palos, CA 93620